



日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 9月 9日
Date of Application:

出願番号 特願2003-317280
Application Number:

[ST. 10/C] : [JP2003-317280]

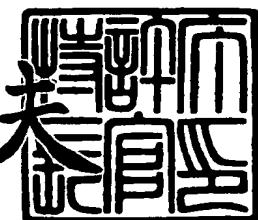
出願人 株式会社リコー
Applicant(s):

日本
特許庁
出願書類
抄本

2003年 9月 24日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



(●)

【書類名】 特許願
【整理番号】 0306269
【提出日】 平成15年 9月 9日
【あて先】 特許庁長官 今井 康夫 殿
【国際特許分類】 G03G 21/00 370
【発明者】
【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内
【氏名】 秋吉 邦洋
【発明者】
【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内
【氏名】 安藤 光男
【発明者】
【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内
【氏名】 田中 浩行
【特許出願人】
【識別番号】 000006747
【氏名又は名称】 株式会社リコー
【代理人】
【識別番号】 100070150
【弁理士】
【氏名又は名称】 伊東 忠彦
【先の出願に基づく優先権主張】
【出願番号】 特願2002-269280
【出願日】 平成14年 9月13日
【手数料の表示】
【予納台帳番号】 002989
【納付金額】 21,000円
【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9911477

【書類名】特許請求の範囲**【請求項 1】**

複数のアプリケーションを搭載可能に構成された画像形成装置において、あるアプリケーションを実行するために必要な必要リソースの情報と、前記画像形成装置において使用できる使用可能リソースの情報を取得する情報取得手段と、前記必要リソースの情報と前記使用可能リソースの情報を比較し、比較の結果に基づき、前記アプリケーションに対する起動制限処理を行う起動制限手段とを備えたことを特徴とする画像形成装置。

【請求項 2】

前記情報取得手段は、前記アプリケーションの実行ファイルに含まれる必要リソース情報を取得する請求項 1 に記載の画像形成装置。

【請求項 3】

前記情報取得手段は、前記アプリケーションを仮起動することにより、前記必要リソース情報を取得する請求項 2 に記載の画像形成装置。

【請求項 4】

前記情報取得手段は、前記必要リソースの情報として、前記アプリケーションが実行された際に使用したリソース量の実績値を取得する請求項 1 に記載の画像形成装置。

【請求項 5】

前記情報取得手段は、前記実績値を、プロセスが使用するリソースに関する情報を保持するシステム情報から取得する請求項 4 に記載の画像形成装置。

【請求項 6】

前記情報取得手段は、前記必要リソースの情報として、前記アプリケーションが使用するメモリ領域容量を取得する請求項 1 ないし 5 のうちいずれか 1 項に記載の画像形成装置。

【請求項 7】

前記メモリ領域容量はヒープ領域容量である請求項 6 に記載の画像形成装置。

【請求項 8】

前記起動制限手段は、前記使用可能リソースのうち前記必要リソースの占める割合を、前記画像形成装置におけるオペレーションパネルに表示する請求項 1 に記載の画像形成装置。

【請求項 9】

前記起動制限手段は、前記起動制限処理として、前記必要リソースの量が前記使用可能リソースの量を超える場合に、前記画像形成装置におけるオペレーションパネルに警告を表示する請求項 1 に記載の画像形成装置。

【請求項 10】

前記起動制限手段は、前記アプリケーションが、前記使用可能リソースとは異なるリソースを、前記使用可能リソースの代わりに使用することができる場合には、アプリケーションに当該異なるリソースを使用させる請求項 9 に記載の画像形成装置。

【請求項 11】

前記必要リソースは、前記アプリケーションが使用する機器であり、前記起動制限手段は、当該機器が前記画像形成装置に備えられていないときに、前記起動制限処理を行う請求項 1 に記載の画像形成装置。

【請求項 12】

画像形成処理で使用されるハードウェア資源と、ハードウェア資源の制御を行うコントロールサービスと、コントロールサービスをサーバとしたクライアントプロセスとして動作し、前記アプリケーションをクライアントとしたサーバプロセスとして動作する仮想アプリケーションサービスを有する請求項 1 ないし 11 のうちいずれか 1 項に記載の画像形成装置。

【請求項 13】

前記仮想アプリケーションサービスは前記情報取得手段及び前記起動制限手段を含む請求

求項12に記載の画像形成装置。

【請求項14】

複数のアプリケーションを搭載可能に構成された画像形成装置におけるアプリケーションの起動制限方法であって、

あるアプリケーションを実行するために必要な必要リソースの情報と、前記画像形成装置において使用できる使用可能リソースの情報とを取得する情報取得ステップと、

前記必要リソースの情報と前記使用可能リソースの情報とを比較し、比較の結果に基づき、前記アプリケーションに対する起動制限処理を行う起動制限ステップとを備えたことを特徴とする起動制限方法。

【請求項15】

前記情報取得ステップにおいて、前記画像形成装置は、前記アプリケーションの実行ファイルに含まれる必要リソース情報を取得する請求項14に記載の起動制限方法。

【請求項16】

前記情報取得ステップにおいて、前記画像形成装置は、前記アプリケーションを仮起動することにより、前記必要リソース情報を取得する請求項15に記載の起動制限方法。

【請求項17】

前記情報取得ステップにおいて、前記画像形成装置は、前記必要リソースの情報として、前記アプリケーションが実行された際に使用したリソース量の実績値を取得する請求項14に記載の起動制限方法。

【請求項18】

前記画像形成装置は、前記実績値を、プロセスが使用するリソースに関する情報を保持するシステム情報から取得する請求項17に記載の起動制限方法。

【請求項19】

前記情報取得ステップにおいて、前記画像形成装置は、前記必要リソースの情報として、前記アプリケーションが使用するメモリ領域容量を取得する請求項14ないし18のうちいづれか1項に記載の起動制限方法。

【請求項20】

前記メモリ領域容量はヒープ領域容量である請求項19に記載の起動制限方法。

【請求項21】

前記起動制限ステップにおいて、前記画像形成装置は、前記使用可能リソースのうち前記必要リソースの占める割合を、前記画像形成装置におけるオペレーションパネルに表示する請求項14に記載の起動制限方法。

【請求項22】

前記起動制限ステップにおいて、前記画像形成装置は、前記起動制限処理として、前記必要リソースの量が前記使用可能リソースの量を超える場合に、前記画像形成装置におけるオペレーションパネルに警告を表示する請求項14に記載の起動制限方法。

【請求項23】

前記起動制限ステップにおいて、前記画像形成装置は、前記アプリケーションが、前記使用可能リソースとは異なるリソースを、前記使用可能リソースの代わりに使用することができる場合には、アプリケーションに当該異なるリソースを使用させる請求項22に記載の起動制限方法。

【請求項24】

前記必要リソースは、前記アプリケーションが使用する機器であり、前記起動制限ステップにおいて、前記画像形成装置は、当該機器が前記画像形成装置に備えられていないときに、前記起動制限処理を行う請求項14に記載の起動制限方法。

【請求項25】

前記画像形成装置は、

画像形成処理で使用されるハードウェア資源と、

ハードウェア資源の制御を行うコントロールサービスと、

コントロールサービスをサーバとしたクライアントプロセスとして動作し、前記アプリ

ケーションをクライアントとしたサーバプロセスとして動作する仮想アプリケーションサービスを有する請求項14ないし24のうちいずれか1項に記載の起動制限方法。

【請求項26】

前記情報取得ステップ及び前記起動制限ステップは、前記仮想アプリケーションサービスにより実行される請求項25に記載の起動制限方法。

【請求項27】

複数のアプリケーションを搭載可能に構成された画像形成装置を、
あるアプリケーションを実行するために必要な必要リソースの情報と、前記画像形成装置において使用できる使用可能リソースの情報とを取得する情報取得手段、

前記必要リソースの情報と前記使用可能リソースの情報とを比較し、比較の結果に基づき、前記アプリケーションに対する起動制限処理を行う起動制限手段、
として機能させるプログラム。

【請求項28】

請求項27に記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。

【書類名】明細書

【発明の名称】画像形成装置およびアプリ起動制限方法

【技術分野】

【0001】

この発明は、コピー、プリンタ、スキャナおよびファクシミリなどの画像形成処理にかかるユーザサービスを提供するアプリケーションのリソース使用量に応じてアプリケーションの起動の制限を行う画像形成装置および起動制限方法に関するものである。

【背景技術】

【0002】

近年では、プリンタ、コピー、ファクシミリ、スキャナなどの各装置の機能を1つの筐体内に収納した画像形成装置（以下、「複合機」という。）が知られている。この複合機は、1つの筐体内に表示部、印刷部および撮像部などを設けるとともに、プリンタ、コピーおよびファクシミリ装置にそれぞれ対応した3種類のソフトウェアを設け、これらのソフトウェアを切り替えることによって、当該装置をプリンタ、コピー、スキャナまたはファクシミリ装置として動作させるものである。

【0003】

このような従来の複合機では、メモリなどの限られたリソース（資源）の範囲内でプリンタ、コピー、ファクシミリ、スキャナなどの各機能単位ですべてのアプリケーションプログラムが起動される。言い換れば、従来の複合機では、すべてのアプリケーションプログラムを起動できる程度のリソースが用意されており、アプリケーションプログラムが起動不可能になる状況が生じる場合は想定されていなかった。

【特許文献1】特開2002-82806号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

ところで、このような従来の複合機では、プリンタ、コピー、スキャナおよびファクシミリ装置に対応するソフトウェアをそれぞれ別個に設けているため、各ソフトウェアの開発に多大の時間を要する。このため、出願人は、表示部、印刷部および撮像部などの画像形成処理で使用されるハードウェア資源を有し、プリンタ、コピーまたはファクシミリなどの各ユーザサービスにそれぞれ固有の処理を行うアプリケーションを複数搭載し、これらのアプリケーションとハードウェア資源との間に介在して、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とするハードウェア資源の管理、実行制御並びに画像形成処理を行う各種コントロールサービスからなるプラットホームを備えた画像形成装置（複合機）を発明した。

【0005】

このような新規な複合機では、アプリケーションの少なくとも2つが共通的に必要とするサービスを提供するコントロールサービスをアプリケーションと別個に設けた構成となっているため、アプリケーションのサイズが従来の複合機で動作するアプリケーションプログラムのサイズに比べて小さなものとなっており、アプリケーションの起動および終了が頻繁に行われる。

【0006】

このため、複合機に搭載されているメモリなどのリソースの使用状況は頻繁に変化し、すべてのアプリケーションを起動した状態とすることができない場合も生じてくる。このような状況でアプリケーションを起動しても、アプリケーションが不正に終了してしまい、複合機の動作が不安定になるという問題がある。

【0007】

また、かかる新規な複合機は、アプリケーションとコントロールサービスとを別個に設けているため、複合機の出荷後にユーザもしくは第三者であるサードパーティが新規なアプリケーションを開発して複合機に搭載可能な構成となっている。このため、かかる複合機では、出荷時に搭載されているコピー、プリンタ、スキャナ、ファクシミリといった画像

形成装置特有のアプリケーション以外に、従来の複合機とは異なるユーザやサードベンダーが開発した新規アプリケーションなども多数起動可能となっている。このように、コピー、プリンタ、ファクシミリ、スキャナなど複合機であらかじめ提供されている特有のアプリケーションは、限りのあるリソースを意識して開発されている。一方、第三者が開発する新規アプリケーションの場合は、開発するベンダーによって必ずしもリソースを意識して開発されていない。従って、リソース使用量のわからない新規アプリケーションを起動させた場合、複合機の動作が不安定となる可能性が高く、さらに、アプリケーションが使用するリソース量が複合機で使用可能な残リソース量を越えた場合は、リソース不足により複合機の動作が停止するという従来の複合機では問題にならなかった新規な課題が生じてくる。

この発明は上記に鑑みてなされたもので、画像形成装置で使用可能なリソース量情報と、起動しようとするアプリケーションが必要とする必要リソース量とに基づいて、アプリケーションを起動制限することを可能とする画像形成装置を提供することを目的とする。

【課題を解決するための手段】

【0008】

上記目的を達成するため、請求項1に記載の発明は、複数のアプリケーションを搭載可能に構成された画像形成装置において、あるアプリケーションを実行するために必要な必要リソースの情報と、前記画像形成装置において使用できる使用可能リソースの情報とを取得する情報取得手段と、前記必要リソースの情報と前記使用可能リソースの情報とを比較し、比較の結果に基づき、前記アプリケーションに対する起動制限処理を行う起動制限手段とを備える。

【0009】

本発明によれば、必要リソース情報と、使用可能リソース情報とを比較して起動制限処理を行うこととしたため、リソースが不足しているにもかかわらずアプリケーションを起動することを防止でき、画像形成装置のシステムの安定性を向上させることができる。

【0010】

請求項2に記載の発明は、請求項1の記載において、前記情報取得手段は、前記アプリケーションの実行ファイルに含まれる必要リソース情報を取得するものである。また、請求項3に記載の発明は、請求項2の記載において、前記情報取得手段は、前記アプリケーションを仮起動することにより、前記必要リソース情報を取得するものである。

【0011】

本発明によれば、アプリケーションの作成時に実行ファイルに含められた必要リソース情報を取得できる。

【0012】

請求項4に記載の発明は、請求項1の記載において、前記情報取得手段は、前記必要リソースの情報として、前記アプリケーションが実行された際に使用したリソース量の実績値を取得する。

本発明によれば、実績値を取得するので、正確な必要リソースの情報を取得できる。

【0013】

請求項5に記載の発明は、請求項4の記載において、前記情報取得手段は、前記実績値を、プロセスが使用するリソースに関する情報を保持するシステム情報から取得するものである。

【0014】

本発明によれば、正確な必要リソースの情報を取得できる。

【0015】

請求項6に記載の発明は、請求項1ないし5のうちいずれか1項の記載において、前記情報取得手段は、前記必要リソースの情報として、前記アプリケーションが使用するメモリ領域容量を取得するものである。

【0016】

本発明によれば、メモリ領域が不足しているか否かのチェックを行うことができる。

【0017】

請求項7に記載の発明は、請求項6の記載において、前記メモリ領域容量はヒープ領域容量であるとするものである。

【0018】

本発明によれば、ヒープ領域が不足しているか否かのチェックを行うことができる。

【0019】

請求項8に記載の発明は、請求項1の記載において、前記起動制限手段は、前記使用可能リソースのうち前記必要リソースの占める割合を、前記画像形成装置におけるオペレーションパネルに表示するものである。

【0020】

本発明によれば、起動しようとするアプリケーションが使用するリソース量が、使用可能リソース量の中で占める割合をユーザが把握することができる。

【0021】

請求項9に記載の発明は、請求項1の記載において、前記起動制限手段は、前記起動制限処理として、前記必要リソースの量が前記使用可能リソースの量を超える場合に、前記画像形成装置におけるオペレーションパネルに警告を表示するものである。本発明によれば、ユーザがリソース不足を知ることができる。

【0022】

請求項10に記載の発明は、請求項9の記載において、前記起動制限手段は、前記アプリケーションが、前記使用可能リソースとは異なるリソースを、前記使用可能リソースの代わりに使用することができる場合には、アプリケーションに当該異なるリソースを使用させるものである。

【0023】

本発明によれば、あるリソースが不足している場合でも、他のリソースを用いることにより、アプリケーションを実行することができる。

【0024】

請求項11に記載の発明は、請求項1の記載において、前記必要リソースは、前記アプリケーションが使用する機器であり、前記起動制限手段は、当該機器が前記画像形成装置に備えられていないときに、前記起動制限処理を行うものである。

【0025】

本発明によれば、例えば、ADFを使用するアプリケーションを、ADFを装着していない画像形成装置で起動しようとしたときに、起動制限処理を行うことができる。

【0026】

請求項12に記載の発明は、請求項1ないし11の記載において、画像形成処理で使用されるハードウェア資源と、ハードウェア資源の制御を行うコントロールサービスと、コントロールサービスをサーバとしたクライアントプロセスとして動作し、前記アプリケーションをクライアントとしたサーバプロセスとして動作する仮想アプリケーションサービスを有するものである。また、請求項13に記載の発明は、請求項12の記載において、前記仮想アプリケーションサービスは前記情報取得手段及び前記起動制限手段を含むものである。

【0027】

請求項14～26に記載の発明は、上記の画像形成装置に適した方法の発明であり、請求項27、28に記載の発明は、上記の画像形成装置に適したプログラム、記録媒体の発明である。これらの発明によても、上記の画像形成装置の発明と同様の作用効果を奏する。

【発明の効果】**【0028】**

本発明によれば、必要リソース情報と、使用可能リソース情報とを比較して起動制限処理を行うこととしたため、リソースが不足しているにもかかわらずアプリケーションを起動することを防止でき、画像形成装置のシステムの安定性を向上させることができる。

【発明を実施するための最良の形態】**【0029】**

以下に添付図面を参照して、この発明にかかる画像形成装置およびアプリ起動制限方法の好適な実施の形態を詳細に説明する。

【0030】**(実施の形態1)**

図1は、この発明の実施の形態1である画像形成装置（以下、「複合機」という）の構成を示すブロック図である。図1に示すように、複合機100は、白黒レーザプリンタ（B&W LP）101と、カラーレーザプリンタ（Color LP）102と、スキャナ、ファクシミリ、ハードディスク、メモリ、ネットワークインターフェースなどのハードウェアリソース103を有するとともに、プラットホーム120とアプリケーション（以下、アプリともいう）130と仮想アプリケーションサービス（VAS：Virtual Application Service）140から構成されるソフトウェア群110とを備えている。

【0031】

一般的のパーソナルコンピュータの場合、多数のアプリを順次起動していくと、リソース不足が生じてストールする可能性がある。一方、本発明のような複合機の場合は、複合機に予めインストールされている複合機特有のコピー、プリンタ、スキャナ、ファクシミリといったアプリに関しては、ストールして利用できなくなることない。

【0032】

しかし、ユーザやサードベンダなどが開発した新規アプリが搭載可能な複合機の場合、複数のアプリケーションが起動設定されている状態で、更に新規のアプリケーションを起動設定した場合、リソース不足が生じることが考えられる。

【0033】

なお、起動設定とは、インストールされているアプリケーションを、複合機の電源投入時に起動するように設定することである。すなわち、本実施の形態の複合機では、アプリケーションをインストールしただけではアプリケーションは起動せず、起動させるには、起動設定をする必要がある。なお、起動設定をすることにより、当該アプリケーションを起動させる旨を示す情報が複合機の記憶装置に記憶される。そして、電源投入時に、その情報が参照され、そのアプリケーションが起動される。

【0034】

なお、起動設定という方法をとらずに、アプリを随时起動できるように複合機を構成することも可能である。

【0035】

複合機100における仮想アプリケーションサービス（VAS）140は、アプリ130とプラットホーム120との間に配置されている。このVAS140は、アプリの使用するリソース使用量に関する使用予定リソース情報（必要リソース情報ともいう）を例えばアプリケーションから取得して、リソース使用情報ファイルをハードディスク（HD）に生成する（後述する図2のHD200）。取得する使用予定リソース情報としては、メモリに確保されるテキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズがある。ここで、テキストメモリ領域とは、各アプリのプログラムがロードされるメモリ領域である。ヒープ領域とは、各アプリが動的に確保するメモリ領域である。スタック領域とは、各アプリが実行されるとき、または各アプリが内部のモジュールを呼び出すときに使用する引数などを格納するために確保される領域である。

【0036】

アプリケーションから使用予定リソース量を取得する場合、アプリケーションの開発時に、アプリケーションの実行ファイル中にアプリケーションが使用するリソースに関する情報を含めておく。そして、VAS140はアプリケーションを、後述する仮起動することにより、プロセス間通信によりアプリケーションからリソースに関する情報を取得する。

【0037】

また、アプリケーションとは別のファイルに当該アプリケーションが使用するリソースに関する情報を記録しておくことにより、V A S 1 4 0 は当該ファイルを参照することによっても使用予定リソース量を取得できる。

【0038】

また、当該アプリが実際に複合機で実行されたことがある場合には、当該アプリのリソース使用実績値を取得することにより、当該アプリが必要とするリソース量を取得できる。

【0039】

また、V A S 1 4 0 は、既に起動設定されているアプリ及びシステムプログラムのリソース使用量と、複合機が有する総リソース量とから、複合機で使用可能なリソース量を取得する。既に起動設定されているアプリ及びシステムプログラムのリソース使用量として、例えば、これらのプログラムが複合機で実行されることにより複合機に記録される実績値を用いることができる。また、既に起動設定されているアプリ及びシステムプログラムのリソース使用量をリソース使用情報ファイルに記録しておき、これを参照するようにしてもよい。このリソース使用量を、複合機の総リソース量から引くことにより、複合機で使用可能なリソース量を求めることができる。

【0040】

そして、V A S 1 4 0 は、あるアプリに対する起動設定時に、当該アプリの使用予定リソース量と、複合機で使用可能なリソース量とを比較する。そして、リソースが不足していれば、起動制限する。すなわち、起動設定しない。また、起動設定しないことの他、アプリが制限起動モードを有している場合には、アプリが制限起動モードで起動されるように設定を行う。なお、制限起動モードとは、例えば、メモリ領域が不足している場合に、メモリの代わりにハードディスクを使用して実行させるモードのことである。

【0041】

また、上記のように、起動設定時ではなく、起動設定されたアプリが起動するときにV A S 1 4 0 がリソースチェックを行い、起動制限を行うことも可能である。

【0042】

プラットホーム120は、アプリケーションからの処理要求を解釈してハードウェア資源の獲得要求を発生させるコントロールサービスと、一または複数のハードウェア資源の管理を行い、コントロールサービスからの獲得要求を調停するシステムリソースマネージャ (SRM) 123と、汎用OS121とを有している。

【0043】

コントロールサービスは、複数のサービスモジュールから形成され、SCS (システムコントロールサービス) 122と、ECS (エンジンコントロールサービス) 124と、MCS (メモリコントロールサービス) 125と、OCS (オペレーションパネルコントロールサービス) 126と、FCS (ファックスコントロールサービス) 127と、NCS (ネットワークコントロールサービス) 128とから構成されている。なお、このプラットホーム120は、あらかじめ定義された関数により前記アプリケーション130から処理要求を受信可能とするアプリケーションプログラムインターフェース (API) を有している。

【0044】

汎用OS121は、UNIX (登録商標) などの汎用オペレーティングシステムであり、プラットホーム120並びにアプリケーション130の各ソフトウェアをそれぞれプロセスとして並列実行する。

【0045】

SRM123のプロセスは、SCS122とともにシステムの制御およびリソースの管理を行うものである。SRM123のプロセスは、スキャナ部やプリンタ部などのエンジン、メモリ、HDDファイル、ホストI/O (セントロI/F、ネットワークI/F、IEEE1394 I/F、RS232C I/Fなど) のハードウェア資源を利用する上

位層からの要求にしたがって調停を行い、実行制御する。

【0046】

具体的には、このSRM123は、要求されたハードウェア資源が使用可能であるか（他の要求により利用されていないかどうか）を判断し、使用可能であれば要求されたハードウェア資源が使用可能である旨を上位層に伝える。また、SRM123は、上位層からの要求に対してハードウェア資源の利用スケジューリングを行い、要求内容（例えば、プリンタエンジンにより紙搬送と作像動作、メモリ確保、ファイル生成など）を直接実施している。

【0047】

SCS122のプロセスは、アプリ管理、操作部制御、システム画面表示、LED表示、リソース管理、割り込みアプリ制御などを行う。

【0048】

ECS124のプロセスは、白黒レーザプリンタ（B&W LP）101、カラーレーザプリンタ（Color LP）102、スキャナ、ファクシミリなどからなるハードウェアリソース103のエンジンの制御を行う。

【0049】

MCS125のプロセスは、画像メモリの取得および解放、ハードディスク装置（HD）の利用、画像データの圧縮および伸張などを行う。

【0050】

FCS127のプロセスは、システムコントローラの各アプリ層からPSTN/ISDN網を利用したファクシミリ送受信、BKM（バックアップSRAM）で管理されている各種ファクシミリデータの登録/引用、ファクシミリ読みとり、ファクシミリ受信印刷、融合送受信を行うためのAPIを提供する。

【0051】

NCS128のプロセスは、ネットワークI/Oを必要とするアプリケーションに対して共通に利用できるサービスを提供するためのプロセスであり、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、アプリケーションからデータをネットワーク側に送信する際の仲介を行う。具体的には、ftpd、httpd、lpd、snmpd、telnetd、smtpdなどのサーバーデーモンや、同プロトコルのクライアント機能などを有している。

【0052】

OCS126のプロセスは、オペレータ（ユーザ）と本体制御間の情報伝達手段となるオペレーションパネル（操作パネル）の制御を行う。OCS126は、オペレーションパネルからキー押下をキーイベントとして取得し、取得したキーに対応したキーイベント関数をSCS122に送信するOCSプロセスの部分と、アプリケーション130またはコントロールサービスからの要求によりオペレーションパネルに各種画面を描画出力する描画関数やその他オペレーションパネルに対する制御を行う関数などがあらかじめ登録されたOCSライブラリの部分とから構成される。このOCSライブラリは、アプリケーション130およびコントロールサービスの各モジュールにリンクされて実装されている。なお、OCS126のすべてをプロセスとして動作させるように構成しても良く、あるいはOCS126のすべてをOCSライブラリとして構成しても良い。

【0053】

アプリケーション130は、ページ記述言語（PDL）、PCLおよびポストスクリプト（PS）を有するプリンタ用のアプリケーションであるプリンタアプリ111と、コピー用アプリケーションであるコピーアプリ112と、ファクシミリ用アプリケーションであるファックスアプリ113と、スキャナ用アプリケーションであるスキャナアプリ114と、ネットワークファイル用アプリケーションであるネットファイルアプリ115と、工程検査用アプリケーションである工程検査アプリ116とを有している。これらの各アプリは、複合機特有（画像形成装置特有）のアプリであり、起動時にVAS140に対して自プロセスのプロセスIDとともにアプリ登録要求メッセージを送信し、アプリ登録要

求メッセージを受信したV A S 1 4 0 によって、起動したアプリに対する登録処理が行われる。

【0054】

アプリケーション130の各プロセス、コントロールサービスの各プロセスは、関数呼び出しとその戻り値送信およびメッセージの送受信によってプロセス間通信を行いながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを実現している。

【0055】

このように、実施の形態1にかかる複合機100には、複数のアプリケーション130および複数のコントロールサービスが存在し、いずれもプロセスとして動作している。そして、これらの各プロセス内部には、一または複数のスレッドが生成されて、スレッド単位の並列実行が行われる。そして、コントロールサービスがアプリケーション130に対し共通サービスを提供しており、このため、これらの多数のプロセスが並列動作、およびスレッドの並列動作を行って互いにプロセス間通信を行って協調動作をしながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを提供するようになっている。また、複合機100には、サードベンダなどの第三者がコントロールサービス層の上のアプリケーション層に新規アプリ117, 118を開発して搭載することが可能となっている。図1では、この新規アプリ117, 118を搭載した例を示している。この新規アプリ117, 118は、複合機特有のアプリ以外のアプリに相当するものである。

【0056】

なお、実施の形態1にかかる複合機100では、複数のアプリケーション130のプロセスと複数のコントロールサービスのプロセスとが動作しているが、アプリケーション130とコントロールサービスのプロセスがそれぞれ単一の構成とすることも可能である。また、各アプリケーション130は、アプリケーションごとに追加または削除することができる。

【0057】

図2に複合機100のハードウェア構成例を示す。

【0058】

複合機100は、コントローラ160と、オペレーションパネル175と、ファックスコントロールユニット（F C U）176と、プリンタ等の画像形成処理に特有のハードウェア資源であるエンジン部177とを含む。コントローラ160は、C P U 1 6 1と、システムメモリ162と、ノースブリッジ（N B）163と、サウスブリッジ（S B）164と、A S I C 1 6 6と、ローカルメモリ167と、H D D 1 6 8と、ネットワークインターフェースカード（N I C）169と、S D カード用スロット170と、U S B デバイス171と、I E E E 1 3 9 4 デバイス172と、セントロニクス173とを含む。なお、メモリ162、167はR A M、R O M等を含む。F C U 1 7 6 およびエンジン部177は、コントローラ160のA S I C 1 6 6にP C I バス178で接続されている。

【0059】

C P U 1 6 1が、複合機100にインストールされるアプリケーション、コントロールサービス等のプログラムを、メモリから読み出して実行する。

【0060】

図3は、実施の形態1にかかる複合機100のV A S 1 4 0 の構成と、V A S 1 4 0 と各アプリ、コントロールサービス層150および汎用O S 1 2 1との関係を示すブロック図である。

【0061】

仮想アプリケーションサービス（V A S）140のプロセスには、ディスパッチャ144と、制御スレッド143と、リソース使用情報取得スレッド141と、アプリ起動制限手段としてのアプリ起動制限スレッド142とが動作している。

【0062】

ディスパッチャ144は、アプリケーション130やコントロールサービスからのメッセージ受信を監視し、受信したメッセージに応じて制御スレッド143、リソース使用情報取得スレッド141、アプリ起動制限スレッド142に対して処理要求を行うものである。

【0063】

アプリケーションを起動する場面において、実施の形態1の複合機100では、ディスパッチャ144は、コントロールサービスからアプリ起動要求メッセージを受信したとき、受信したアプリ起動要求メッセージを制御スレッド143に送信し、制御スレッド143からリソース使用情報取得スレッド141とアプリ起動制限スレッド142に対して処理要求を行う。

【0064】

また、起動設定の場面でも、例えば、ディスパッチャ144が処理要求メッセージを制御スレッド143に送信し、制御スレッド143からリソース使用情報取得スレッド141とアプリ起動制限スレッド142に対して処理要求を行う。

【0065】

制御スレッド143は、ディスパッチャ144からの処理要求メッセージを受信して、リソース使用情報取得処理およびアプリ起動制限処理を行う。ここで、アプリ起動制限処理とは、必ずアプリ起動制限を行うものではなく、起動するアプリのリソース使用量に応じて、アプリを制限的に起動する場合も含む。

【0066】

また、制御スレッド143は、HD200に格納されたリソース使用情報ファイル201を参照することにより、インストールされたアプリの中でリソース使用量に関するリソース使用情報が記録されていないアプリの有無を判別することができる。そして、制御スレッド143は、リソース使用情報取得スレッドからリソース使用情報を取得して、アプリ起動制限スレッド142に送信する。なお、アプリ起動制限スレッド142が、リソース使用情報ファイル201を参照してもよい。アプリ起動制限スレッド142では、複合機の総リソース量から既に起動設定されているアプリが使用するリソースを除いた使用可能なリソース量情報と、これから起動設定、もしくは、これから起動しようとする新規アプリが使用する予定の使用予定リソース量とを比較して、使用予定リソース量の方が使用可能なリソース量を越えている場合は、起動制限処理を行う。

【0067】

リソース使用情報取得スレッド141は、制御スレッド143から処理要求を受けると、汎用OS121が管理するRAM210上のproc構造体211（またはu領域212）を参照して、起動しようとするアプリが使用するテキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズを取得して、リソース使用情報ファイル201としてハードディスク(HD)200に生成する。かかるリソース使用情報は、アプリごとのレコードとして記録される。また、前述したように、リソース使用情報取得スレッド141は、アプリの実行ファイル中に含まれる当該アプリの使用予定リソース情報を取得し、それをリソース使用情報ファイル201に記録してもよい。

【0068】

また、リソース使用情報取得スレッド141は、複合機における使用可能リソース量を算出するために必要な、現在のプロセスが使用しているリソース量もproc構造体211から取得できる。

【0069】

上記のように各スレッドが処理を行う代わりに、VAS140が1プロセスで処理を行うようにしてもよい。

【0070】

上記VAS140のプログラムは、ソフトウェア開発キット(SDK: Software Development Kit)等の一部または全部として、CD-ROMまたはFD(フレキシブルディスク)などの記憶媒体に実行可能な形式またはインストール可能

な形式のファイルで提供される。また、このような実行可能な形式またはインストール可能な形式のVAS140のプログラムファイルを、ネットワーク経由で取得可能な方法で提供するようにしても良い。

【0071】

図4は、HD200に格納されるリソース使用情報ファイル201の内容例を示す説明図である。図4に示すように、リソース使用情報ファイル201には、アプリID（プロダクトIDともいう）ごとに、テキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズなどが記録されている。

【0072】

なお、このファイルにアプリが使用する機器（ADFなど）を記録しておいてもよい。これにより、複合機における実際の機器構成により起動制限を行うことも可能になる。

【0073】

図5は、リソース使用情報取得スレッド141が参照するproc構造体211の一例を示す説明図である。図5に示すように、proc構造体211には、各プロセスごとに、プロセスID(p_pid)、テキストメモリ領域サイズ、ヒープメモリ領域サイズ、スタック領域サイズなどが格納されている。このproc構造体211は、プロセス実行時、プロセス終了時、およびプロセスの状態が変化したときに汎用OS121によって更新される。

【0074】

また、リソース使用情報取得スレッド141は、汎用OS121のシステムコールあるいはコントロールサービスで提供されるサービス関数呼び出しによって、proc構造体211から、既に起動設定されているアプリやシステムが使用するリソース量を取得し、総リソース量からそのリソース量を除いた使用可能な残リソース量（例えば、テキストメモリ領域の残容量、ヒープ領域の残容量、スタック領域など）を取得する。

【0075】

アプリ起動制限スレッド142は、コントロールサービス層150から新規アプリの起動要求があった場合、もしくは起動設定要求があった場合、リソース使用情報ファイル201を参照して、その新規アプリが使用するリソース使用量に関する情報を取得する。また、アプリ起動制限スレッド142は、前記の残リソース量を、例えば、リソース使用情報取得スレッド141から取得する。そして、両者を比較して、アプリの起動を制限する必要があるか否かを判断する。

【0076】

起動設定においては、アプリ起動制限スレッド142は、アプリ起動制限が必要と判断すると、画面に後述するような警告を表示する。

【0077】

また、実際の起動の場面においては、アプリ起動制限スレッド142は、アプリ起動制限が必要と判断すると、アプリ起動要求のあったアプリに対して起動制限要求メッセージを送信する。一方、アプリ起動制限をすべきでないと判断した場合は、アプリ起動要求のあったアプリに対して起動要求メッセージを送信して、アプリの通常起動処理が行われる。

【0078】

図6は、複合機のオペレーションパネル上にユーザアプリの起動制限レベルをヒストグラムで表示した図である。この図は、起動設定時に表示されるものである。図6に表示されるヒストグラムは、コピーアプリなどの複合機特有のアプリやシステムプログラムが使用するリソースを除いた使用可能な残リソース量のみを示したものである。すなわち、残リソースを複数のユーザアプリ（新規アプリ）でどのように分配して利用すれば良いかをユーザが視覚的に判断できるようにしたものである。ヒストグラムの内訳は、1番目に起動設定したユーザアプリ221のリソース使用量と、2番目に起動設定したユーザアプリ222のリソース使用量とがそれぞれ表示され、これから起動設定しようとする3番目のユーザアプリ223の場合は、予想リソース使用量を示している。図6の例では、3番目

のユーザアプリ223を起動すると残リソース量が危険レベルに達するので、起動設定できない。ここでは、残リソース量をヒストグラムで表すだけでなく、起動制限レベルを安全（青）、注意（黄）、危険（赤）の3段階に色分け表示して、さらに視認効果を向上させている。なお、これ以外にも、危険レベルになると表示を点滅させたり、音声ガイダンスや警告音を発するようにしても勿論良い。

【0079】

上記のようなヒストグラムを含む、起動設定時にオペレーションパネルに表示される画面の例を次に示す。なお、これから説明する画面は、例えば、VAS140が表示を行う。

【0080】

図7は、「簡易スキャナ」と「簡易コピー」が既に起動設定されている場合の起動設定画面を示す図である。同図に示すように、「簡易スキャナ」と「簡易コピー」が使用するメモリ消費量が右側のヒストグラム251に表示される。

【0081】

ここで、次に「簡易プリント」を起動設定するためにこれを選択すると、図8に示すように、「簡易プリント」の部分が数秒間ブリンクする。そして、メモリ消費量は100パーセントとなり、これもブリンクする。

【0082】

その後、図9又は図10に示す画面が表示される。図9は、他のアプリケーションの選択を促す画面である。OKを押すことにより、図7の画面に戻る。図10は、メモリ不足でも起動設定を可能とする画面の例である。なお、キーが割り付けられていないアプリケーションを選択した場合には、図11に示す画面が表示される。

【0083】

図7の状態から、「簡易スキャナ」を選択し、起動設定が完了すると、図12の画面を経て、図13に示す画面が表示される。図13に示すように、「簡易スキャナ」の予定メモリ消費量の分だけ、ヒストグラムの表示値が増加（252）する。

【0084】

あるアプリの起動設定を解除する際には図14～図18に示す画面が表示される。

【0085】

図14の状態から、起動設定解除しようとする「簡易スキャナ」を指定し、OKを押すと、図15に示すように、「簡易スキャナ」がブリンク表示される。ここでOKを押すことにより、図16の画面が表示され、OKを押すことにより、図17の画面を経て図18の画面が表示される。図18に示すように、「簡易スキャナ」の分のメモリ消費量の分がヒストグラムから減少する。

【0086】

次に、複合機100のVAS140によるアプリ起動制限処理について図19のフローチャートを参照して説明する。図19に示す処理は、起動設定時におけるVAS140による処理を示すフローチャートである。

【0087】

まず、VAS140があるアプリに対する起動設定要求を受信すると（ステップS601）、VAS140は、そのアプリに対するリソース使用情報ファイルが作成されているか否かをHD200を参照して判断する（ステップS602）。

【0088】

リソース使用情報ファイルが作成されている場合、VAS140は、リソース使用情報をリソース使用情報ファイル201から取得する（ステップS603）。また、上記ステップS602で、新規アプリのリソース使用情報ファイルが作成されていなかった場合は、VAS140は、リソース使用情報ファイルを作成する（ステップS604）。これは、例えば、制御スレッド143が、リソース使用情報取得スレッド141に対してリソース使用情報ファイルを作成するように処理要求することにより、リソース使用情報取得スレッド141が作成を行う。

【0089】

ステップS604のリソース使用情報ファイルの作成処理の詳細なサブルーチンは、後述する図21のフローチャートを用いて説明する。このリソース使用情報ファイルが作成処理されると、上記したステップS603の処理が行われる。

【0090】

続いて、VAS140は、既に起動設定されているアプリおよびシステムプログラムが使用するリソースを除いた使用可能なリソース量情報を取得する（ステップS605）。既に起動設定されているアプリおよびシステムプログラムが使用するリソースは、汎用OS121のシステムコールあるいはコントロールサービス層150で提供されるサービス関数呼び出しを用いることにより、proc構造体から取得できる。なお、既に起動設定されているアプリのうち、リソース使用情報ファイルにリソース情報を有するものに対しては、リソース使用情報ファイルからリソース情報を取得してもよい。

【0091】

続いて、VAS140は、起動設定しようとするアプリが使用する予定リソース量に関する情報と、使用可能なリソース量情報を比較する（ステップS606）。ここで、アプリ起動制限スレッド142は、使用予定リソース量が使用可能リソース量を越えているか否かを判断し（ステップS607）、越えている場合は、図9に示したような画面を表示して、ユーザに警告を行う（ステップS608）。越えていない場合、通常のアプリ起動設定処理が行われる（ステップS609）。

【0092】

なお、起動設定しようとするアプリケーションが、メモリの代わりにハードディスクを使用して実行できるモードを有している場合には、図9に示した警告に代えて、そのモードを使用するか否かをユーザに尋ねる画面を表示してもよい。ユーザがそのモードを使用することを選択した場合には、起動時にそのモードでアプリが起動される。なお、そのモードを有していることは、VAS140がアプリ情報を取得するときに把握し、その旨をリソース使用情報ファイルに記録しておくことができる。

【0093】

図20は、起動設定したアプリを起動する場合に、起動制限処理を行う場合のフローチャートである。

【0094】

まず、ディスパッチャ144がコントロールサービス層150からアプリ起動要求メッセージを受信すると、アプリ起動要求メッセージをそのアプリのプロセスIDとともに制御スレッド143に受け渡す。制御スレッド143は、アプリ起動要求メッセージとプロセスIDとをディスパッチャ144から受信すると、起動要求のあったアプリが複合機特有のアプリか否かを判断する（ステップS701）。複合機特有のアプリ以外のアプリの場合、すなわち、新規アプリに対する起動要求があった場合（ステップS702）、制御スレッド143は、そのアプリのリソース使用情報ファイルが作成されているか否かをHD200を参照して判断する（ステップS703）。

【0095】

リソース使用情報ファイルが作成されている場合、制御スレッド143は、起動要求のあった新規アプリのリソース使用情報をリソース使用情報ファイル201からアプリ起動制限スレッド142へ送る（ステップS604）。また、上記ステップS703で、新規アプリのリソース使用情報ファイルが作成されていなかった場合は、制御スレッド143は、リソース使用情報取得スレッド141に対してリソース使用情報ファイルを作成するように処理要求する（ステップS705）。このリソース使用情報ファイルが作成処理されると、上記したステップS704の処理が行われる。

【0096】

続いて、VAS140は、図19と同様に、使用可能な残リソース量を取得する（ステップS706）。

【0097】

また、アプリ起動制限スレッド142では、起動要求のあった新規アプリが使用する使用予定リソース量に関する情報と、複合機で使用可能なリソース量情報とを比較する（ステップS707）。

【0098】

ここで、アプリ起動制限スレッド142は、リソース使用量がリソース量を越えているか否かを判断し（ステップS708）、越えている場合、画面に警告を表示して当該アプリの起動を制限する。（ステップS709）。また、新規アプリが起動制限モードを有している場合には、そのモードを使用するか否かをユーザに問い合わせる画面を表示し、そのモードで実行することが許可された場合にのみ、そのモードでアプリを起動するようにしてもよい。使用予定リソース量が使用可能リソース量を越えていない場合、アプリ起動制限処理は行わず、通常のアプリ起動処理が行われる（ステップS710）。このステップS710の通常のアプリ起動処理は、図20の最初のステップS601において、複合機特有のアプリに対する起動要求があった場合も同様に行われる。本発明において、複合機特有のアプリの場合は、使用するリソースがあらかじめ確保されているため、何時起動処理を行ってもリソース不足が生じない。

【0099】

図21は、リソース使用情報ファイルの作成処理の動作を説明するフローチャートである。

【0100】

まず、リソース使用情報を得たいアプリをVAS140が仮起動させる（ステップS801）。そして、プロセス間通信によりリソース情報をアプリから取得する（ステップS802）。仮起動させることにより得たリソース情報に基づき、HD200内のリソース使用情報ファイル201に各アプリ単位でリソース使用情報ファイルを作成する（ステップS803）。図4は、作成されたリソース使用情報ファイルの一例を示したものであるが、アプリIDごとに、その使用リソースとして、テキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズごとにリソース量が格納されている。

【0101】

また、起動設定しようとするアプリが既に実行されたことがあり、リソース使用量の実績値がproc構造体211にある場合には、アプリからリソース情報を取得することに代えて、当該実績値を取得してもよい。この実績値を用いて、リソース使用情報ファイルをアプリが起動されるたびに更新すれば、学習機能を持たせたのと同様に常に正確なリソース使用情報を得ることができる。なお、取得する度に実績リソース量が変動する場合は、実績値の平均値、あるいは、最大値（MAX）を、リソース使用情報ファイルに記録するリソース情報として使用できる。

【0102】

このRAM210のproc構造体211のリソース使用情報を使ってリソース使用情報ファイルを作成する場合は、リソース使用情報取得スレッド141がアプリID、プロセスIDとリソース使用情報取得要求メッセージを制御スレッド143から受信すると、RAM210のproc構造体211を参照して該当するアプリIDのプロセスIDのブロックの位置を検索する。そして、検索されたプロセスIDのブロックから、テキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズのリソース使用情報を取得し、アプリIDとともにリソース使用情報ファイル201に記録する。

【0103】

次に、アプリケーションの仮起動について説明する。仮起動は、複合機のリソースを使用することになるアプリの通常起動（アプリ本来の機能を奏するための起動を通常起動と呼ぶ）とは別の起動である。仮起動では、アプリはアプリ本来の動作に必要なメモリ確保等のリソース取得を行わず、VAS140とのプロセス間通信処理のみを行う。そして、アプリは、リソース情報をVAS140に提供する。仮起動したアプリのプロセスは、VAS140との通信処理が終了すれば終了する。また、アプリの仮起動に関する機能は、アプリ本来の機能によらず、本実施の形態における複合機100で動作するアプリに共通

する機能である。従って、例えば、ベンダーがアプリを開発する場合、ベンダーに、仮起動の機能を含むプログラムテンプレートを提供し、そのプログラムテンプレートを用いてベンダーが複合機用のアプリを開発することができる。なお、アプリにリソース情報を含めるには、例えば、ベンダーが、使用予定リソース情報を記録したインクルードファイルを作成し、アプリのコンパイル時にインクルードする。

【0104】

図22に、仮起動の機能を含むアプリのプログラム記述（メイン関数）の概要を示す。なお、この記述を上記のプログラムテンプレートとして提供する。

【0105】

図22に示すように、このプログラム記述は、アプリケーションを仮起動するか通常起動するかを引数（-v）によって指定する。これにより、VASがアプリを起動する際に、通常起動と仮起動とを容易に使い分けることができる。すなわち、引数（-v）を使って仮起動を指定すると、仮起動が実行され、アプリ情報提供処理がなされる。また、仮起動が指定されていない場合は、通常起動を行って、アプリ本来の動作を行う。

【0106】

このように、実施の形態1にかかる複合機100では、アプリが使用するリソース使用量に関する情報を取得してリソース使用情報ファイル201を生成し、複合機で使用可能なリソース量と、起動設定要求された新規アプリの使用予定リソース量とを比較して、使用予定リソース量が使用可能リソース量を越えている場合は、新規アプリの起動を制限するので、リソース不足により動作が不安定になったり、ストールする心配がなくなり、ユーザやサードベンダが開発した新規アプリを自由に搭載して起動させることができる。

【0107】

また、実施の形態1にかかる複合機100では、新規アプリのリソース使用状況によって複合機特有のアプリの実行ができなくなることを未然に回避して、複合機100のシステムの安定性を向上させることができる。

【0108】

なお、実施の形態1にかかる複合機100では、VAS140が全てのアプリケーション130に対してリソース使用情報取得処理、アプリ起動制限処理を行っているが、一部のアプリに対してのみかかる処理を行うように構成しても良い。例えば、新規アプリ117、118などサードベンダなどの第三者が開発したアプリにのみリソース使用情報取得処理、アプリ起動制限処理を行い、プリンタアプリ111やコピーアプリ112などの既存のアプリに対してはこのようなサービスを行わないように構成しても良い。

【0109】

また、新規アプリのリソース使用量が変動することを考えると、使用予定リソース量が使用可能リソース量を越えないまでも、一定の範囲内にある場合は、新規アプリの起動を制限するようにしてもよい。

【0110】

（実施の形態2）

実施の形態1にかかる複合機100は、VAS140が全アプリケーションに対して1つのみ存在するものであったが、この実施の形態2にかかる複合機では、各アプリごとに一つのVASが起動し、各VASは対応するアプリに対してのみリソース使用情報取得およびアプリ起動制限を行うものである。

【0111】

図23は、実施の形態2にかかる複合機800の構成を示すブロック図である。図23に示すように、複合機800では、複数の仮想アプリケーションサービス（VAS）841～848がアプリケーション130の各アプリごとに動作している点が、実施の形態1にかかる複合機100と異なっている。

【0112】

VAS841～848は、プリンタアプリ111、コピーアプリ112、ファックスアプリ113、スキャナアプリ114、ネットファイルアプリ115、工程検査アプリ11

6、新規アプリ117および118に対応して、リソース使用情報取得処理およびアプリ起動制限処理を行う。

【0113】

図24は、実施の形態2にかかる複合機800のVAS841～848の構成と、VAS841～848と各アプリ、コントロールサービス層150および汎用OS121との関係を示すブロック図である。

【0114】

また、実施の形態2にかかる複合機800では、実施の形態1の複合機100と異なり、図24に示すように、各VAS841～848と各アプリとの間にはVAS制御プロセス（デーモン）801が動作している。

【0115】

このVAS制御プロセス（デーモン）801は、各アプリに対応したVAS841～848を生成する。また、VAS制御プロセス801は、HD200に格納されたリソース使用情報ファイル201を参照して、起動要求もしくは起動設定要求が行われたアプリについて、リソース使用情報ファイル201にリソース使用情報が記録されているか否かをチェックして、リソース使用情報ファイルを作成する必要があるか否かを判断する。そして、アプリのリソース使用情報が記録されていない場合は、リソース使用情報取得スレッド141がリソース使用情報を取得して、アプリ起動制限スレッド142に送るとともに、リソース使用情報ファイル201を作成する。また、アプリのリソース使用情報が記録されている場合は、リソース使用情報ファイル201からリソース使用情報を読み出してアプリ起動制限スレッド142に送る。

【0116】

仮想アプリケーションサービス（VAS）841～848のプロセスには、ディスパッチャ144と、リソース使用情報取得スレッド141と、アプリ起動制限スレッド142とが動作している。これらの各スレッドの機能は、実施の形態1における対応するスレッドの機能と同様である。

【0117】

このように実施の形態2にかかる複合機800によれば、実施の形態1にかかる複合機100と同様に、複合機800のシステムの安定性を向上させることができる。

【0118】

また、実施の形態2にかかる複合機800では、VAS841～848は起動されるアプリケーション130ごとに別個に起動されるので、複数のアプリケーション130に対する起動制限処理を各アプリケーション130に対応したVAS841～848によって並列に実行できることから、起動アプリケーションに対する起動制限処理を効率的に行うことができる。

【0119】

なお、実施の形態2にかかる複合機800では、全てのアプリごとに別個にVAS841～848を起動していたが、一部のアプリに対してのみVASを起動するように構成しても良い。例えば、新規アプリ117、118などサードベンダなどの第三者が開発したアプリに対してのみVAS847、848を起動してリソース使用情報の取得処理やアプリ起動制限処理を行い、プリンタアプリ111やコピーアプリ112などの既存のアプリに対してはこのようなサービスを行わないように構成することができる。

【0120】

また、実施の形態1および2にかかる複合機100、800では、リソースとして、テキストメモリ領域、ヒープ領域、スタック領域などを利用してリソース使用情報取得およびアプリ起動制限を行っていたが、かかるリソースは一例であって、これ以外のリソースを利用した構成としても勿論良い。

【図面の簡単な説明】

【0121】

【図1】実施の形態1にかかる複合機の構成を示すブロック図である。

【図2】実施の形態1にかかる複合機のハードウェア構成図である。

【図3】実施の形態1にかかる複合機のV A Sの構成と、V A Sと各アプリ、コントロールサービス層および汎用OSとの関係を示すブロック図である。

【図4】実施の形態1にかかる複合機におけるリソース使用情報ファイルの内容例を示す説明図である。

【図5】実施の形態1にかかる複合機におけるV A Sのリソース使用情報取得スレッドが参照するp r o c構造体の一例を示す説明図である。

【図6】複合機のオペレーションパネル上にユーザアプリの起動制限レベルをヒストグラムで表示した図である。

【図7】「簡易スキャナ」と「簡易コピー」が既に起動設定されている場合の起動設定画面を示す図である。

【図8】「簡易プリント」を選択した場合の画面である。

【図9】他のアプリケーションの選択を促す画面である。

【図10】メモリ不足でも起動設定を可能とする画面の例である。

【図11】キーが割り付けられていないアプリケーションを選択した場合の画面である。

【図12】起動設定が完了したことを示す画面である。

【図13】「簡易スキャナ」の起動設定が完了した後の画面である。

【図14】起動設定済みのアプリケーションを示す画面である。

【図15】「簡易スキャナ」を選択した場合の画面である。

【図16】起動設定解除を確認する画面である。

【図17】解除が完了したことを示す画面である。

【図18】解除後の画面である。

【図19】起動設定時における起動制限処理手順を示すフローチャートである。

【図20】起動時における起動制限処理手順を示すフローチャートである。

【図21】リソース使用情報ファイルの作成処理のフローチャートである。

【図22】仮起動の機能を含むアプリのプログラム記述（メイン関数）の概要を示す図である。

【図23】実施の形態2にかかる複合機の構成を示すブロック図である。

【図24】実施の形態2にかかる複合機のV A Sの構成と、V A Sと各アプリ、コントロールサービス層および汎用OSとの関係を示すブロック図である。

【符号の説明】

【0122】

100 複合機

101 白黒レーザプリンタ

102 カラーレーザプリンタ

103 ハードウェアリソース

110 ソフトウェア群

111 プリンタアプリ

112 コピーアプリ

113 ファックスアプリ

114 スキャナアプリ

115 ネットファイルアプリ

116 工程検査アプリ

117, 118 新規アプリ

120 プラットホーム

121 汎用OS

122 SCS

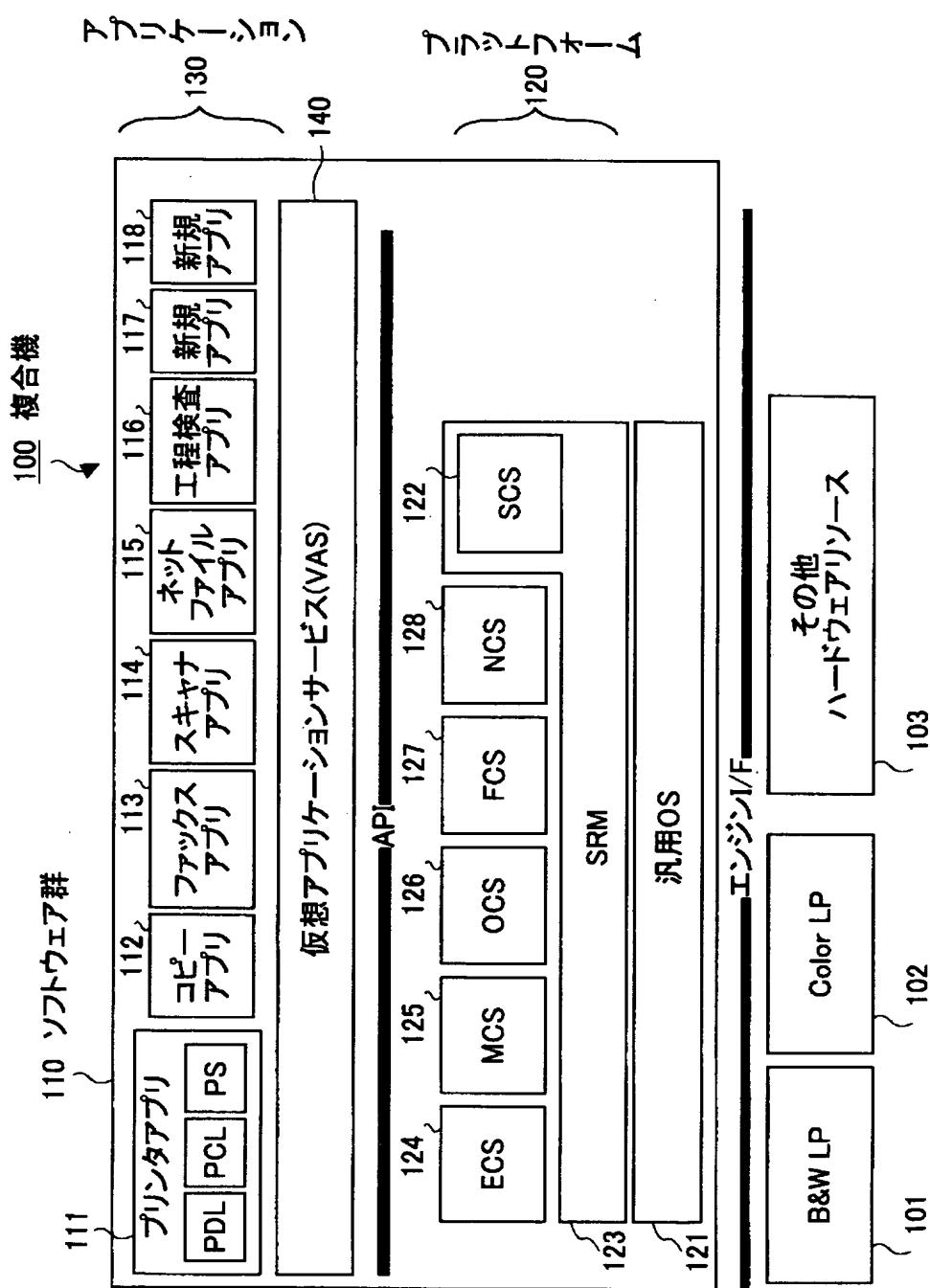
123 S RM

124 E CS

125 MCS
126 OCS
127 FCS
128 NCS
130 アプリケーション
140, 841～848 仮想アプリケーションサービス (V A S)
141 リソース使用情報取得スレッド
142 アプリ起動制限スレッド
143 制御スレッド
144 ディスパッチャ
150 コントロールサービス層
200 ハードディスク (H D)
201 リソース使用情報ファイル
210 R A M
211 p r o c 構造体
212 u 領域
221 ユーザアプリ
222 ユーザアプリ
223 ユーザアプリ
800 複合機
801 V A S 制御プロセス (デーモン)

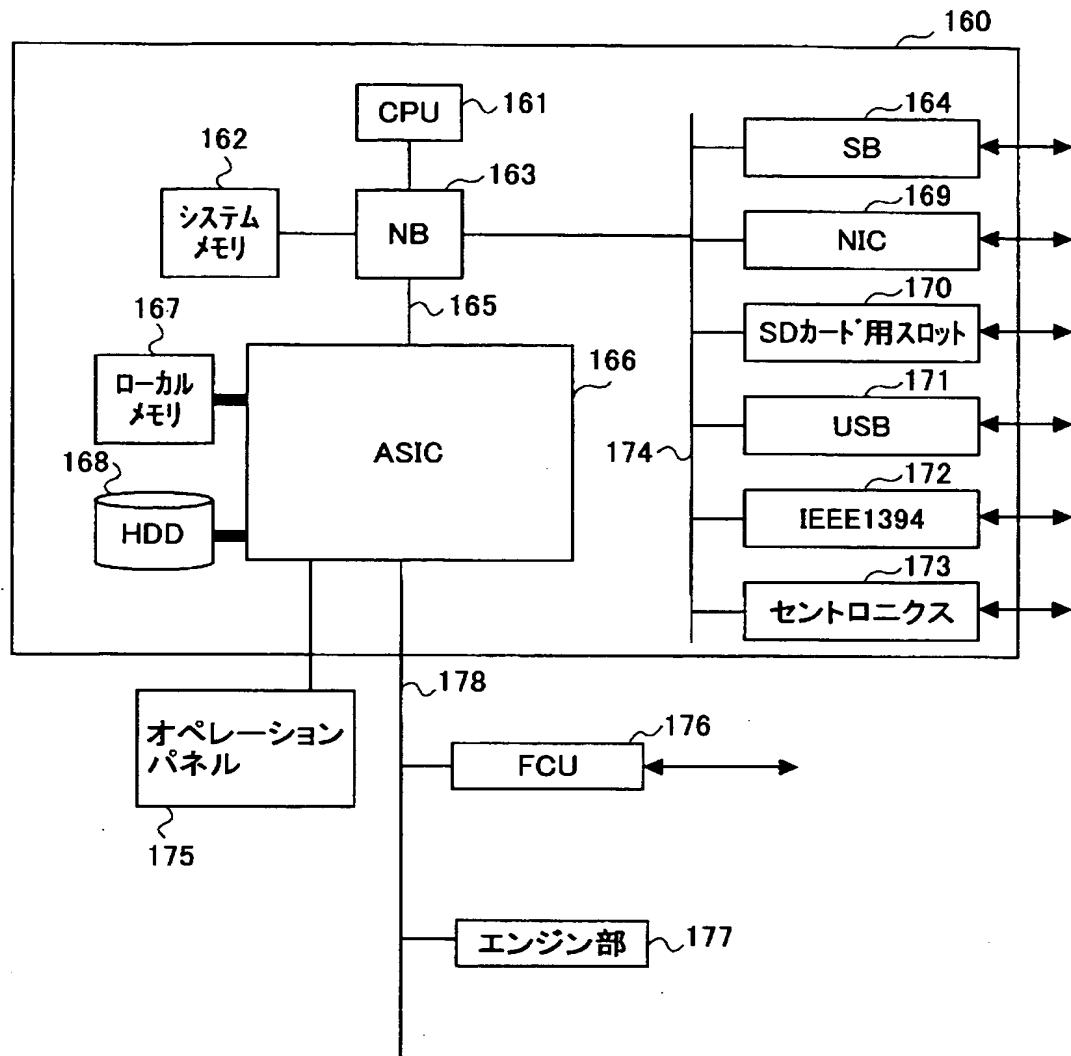
【書類名】 図面
【図 1】

実施の形態 1 にかかる複合機の構成を示すブロック図



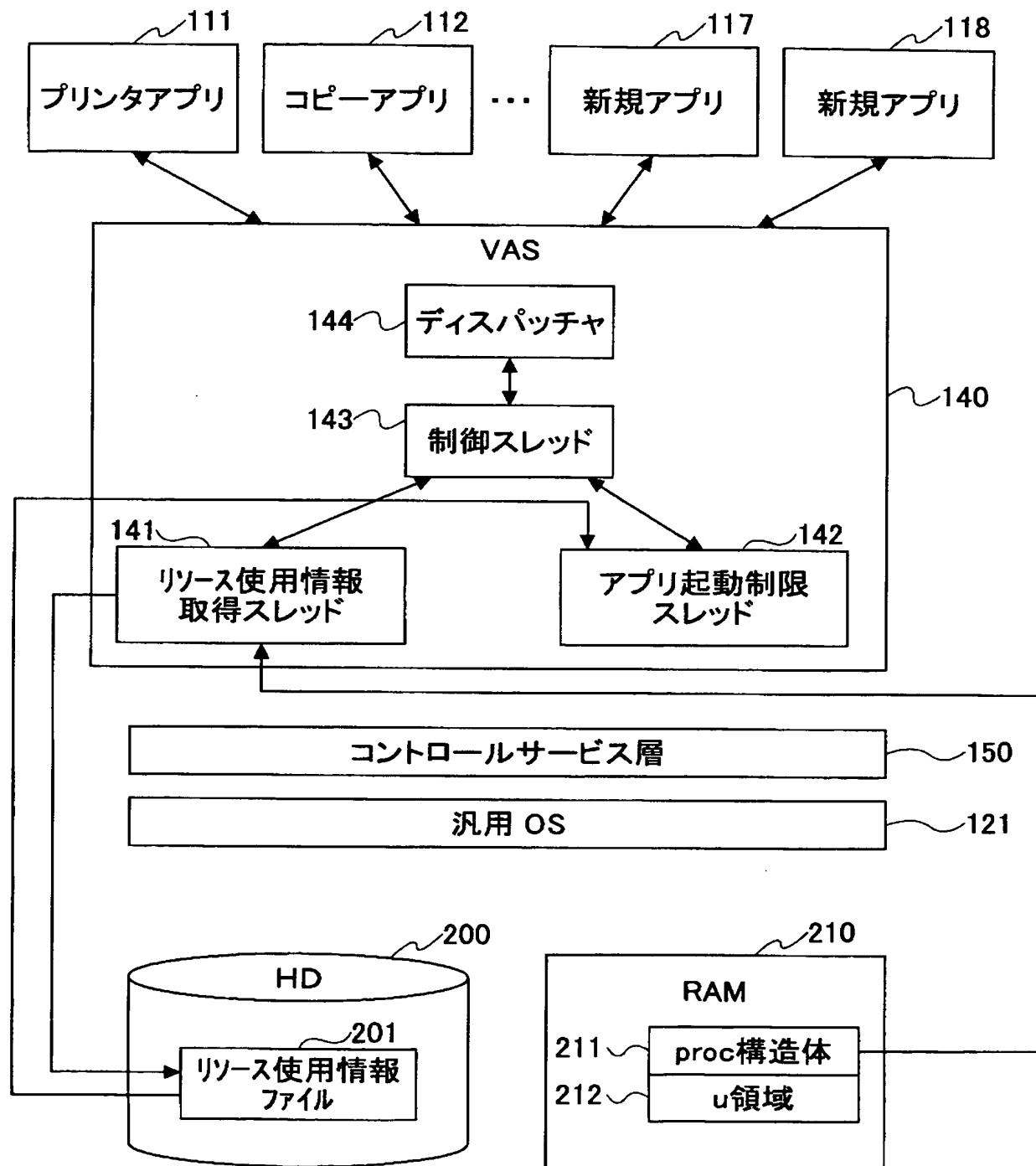
【図2】

実施の形態1にかかる複合機のハードウェア構成図



【図 3】

実施の形態 1 にかかる複合機の VAS の構成と、VAS と各アプリ、コントロールサービス層および汎用 OS との関係を示すブロック図



【図 4】

実施の形態 1 にかかる複合機におけるリソース
使用情報ファイルの内容例を示す説明図

201

リソース使用情報ファイル	アプリID	テキストメモリ 領域サイズ	ヒープ領域 サイズ	スタック領域 サイズ
	101	3MB	1MB	10KB
	102	2MB	500KB	5KB
	103	4MB	200KB	3KB
	⋮	⋮	⋮	⋮
	⋮	⋮	⋮	⋮
	⋮	⋮	⋮	⋮

【図 5】

実施の形態 1 にかかる複合機における V A S のリソース使用情報
取得スレッドが参照する p_r_o_c 構造体の一例を示す説明図

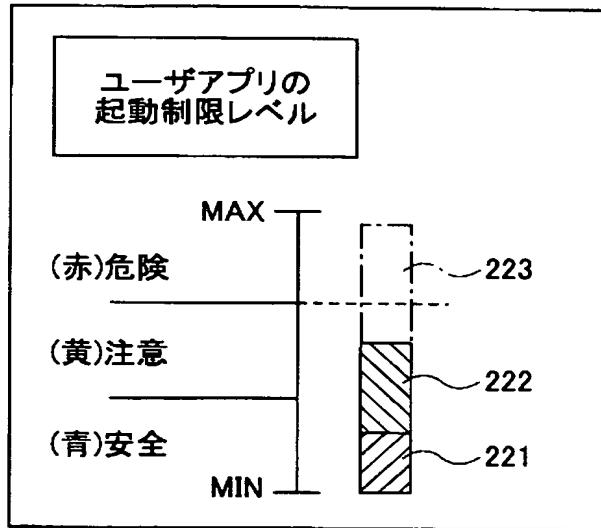
211

proc構造体	
メンバ名 p_pid	説明 プロセスID
p_txtsz	テキストメモリ領域サイズ
p_heapsz	ヒープ領域サイズ
p_stacksz	スタック領域サイズ
⋮	⋮
⋮	⋮

【図 6】

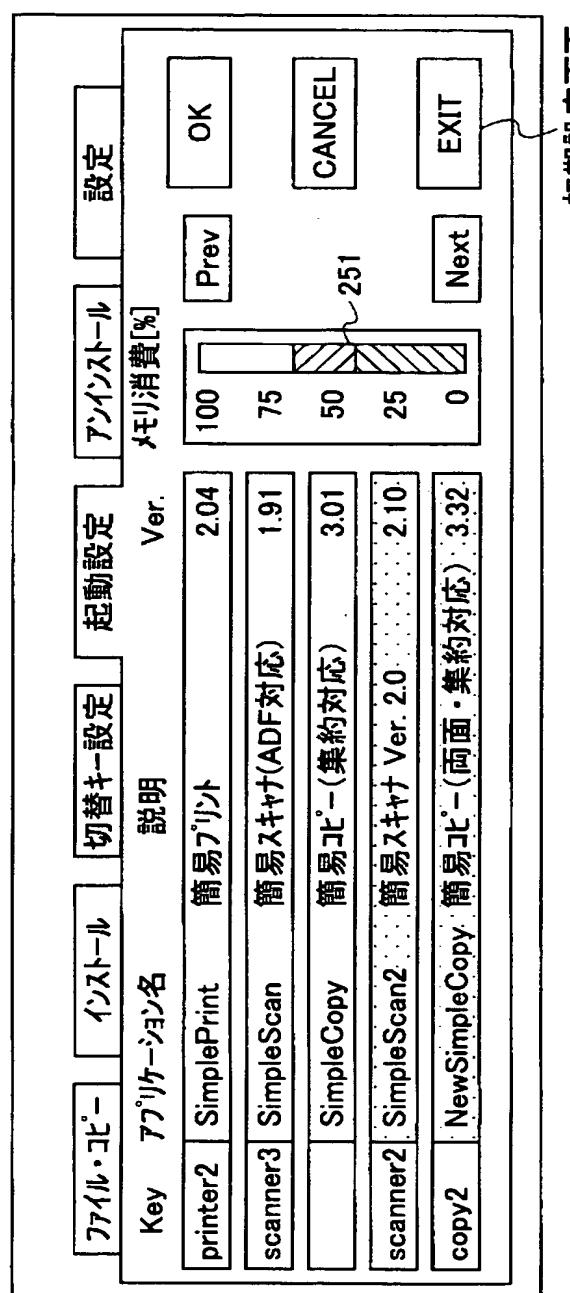
複合機のオペレーションパネル上にユーザアプリの起動制限レベルをヒストグラムで表示した図

オペレーションパネル(ヒストグラム)



【図 7】

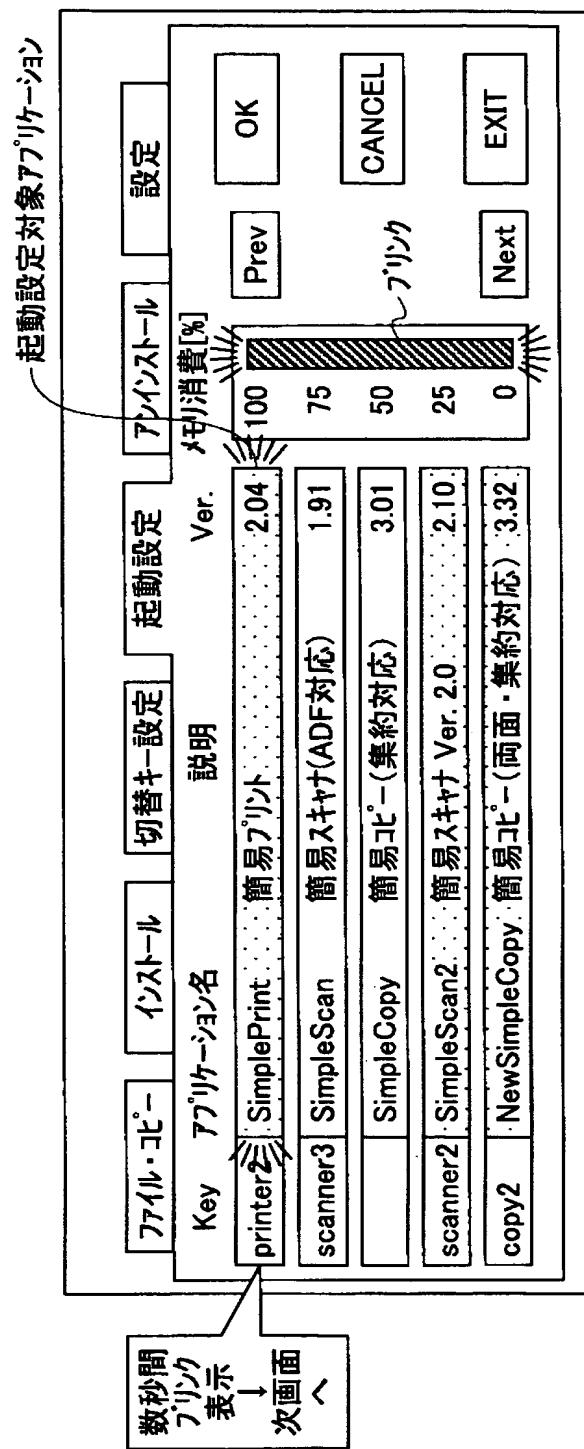
「簡易スキャナ」と「簡易コピー」が既に起動設定されている場合の起動設定画面を示す図



初期設定画面へ

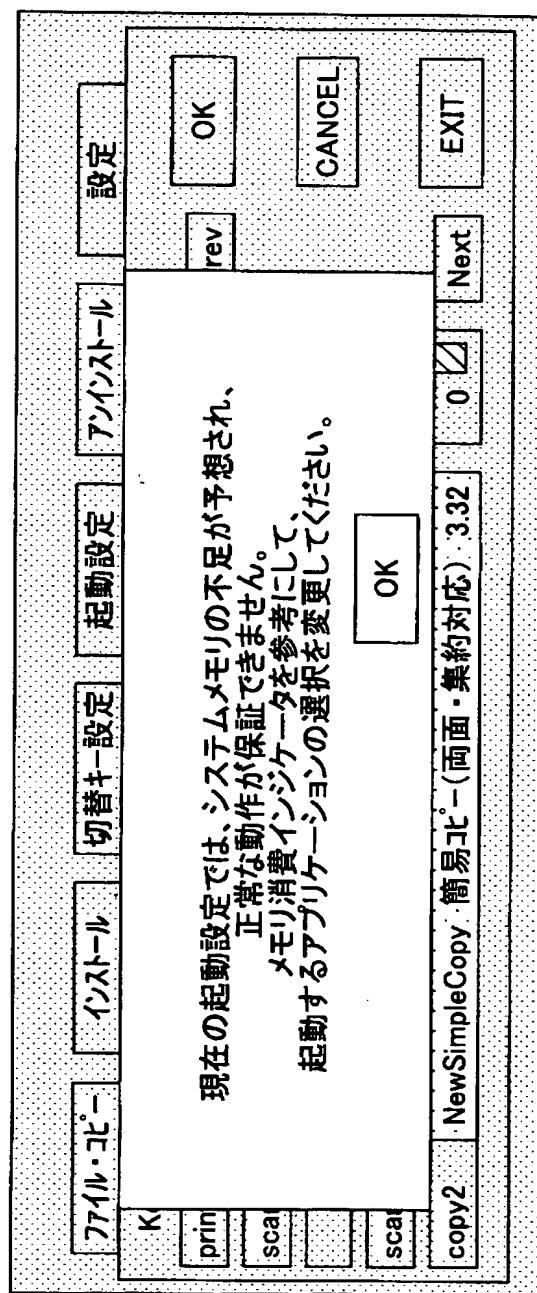
【図8】

「簡易プリント」を選択した場合の画面



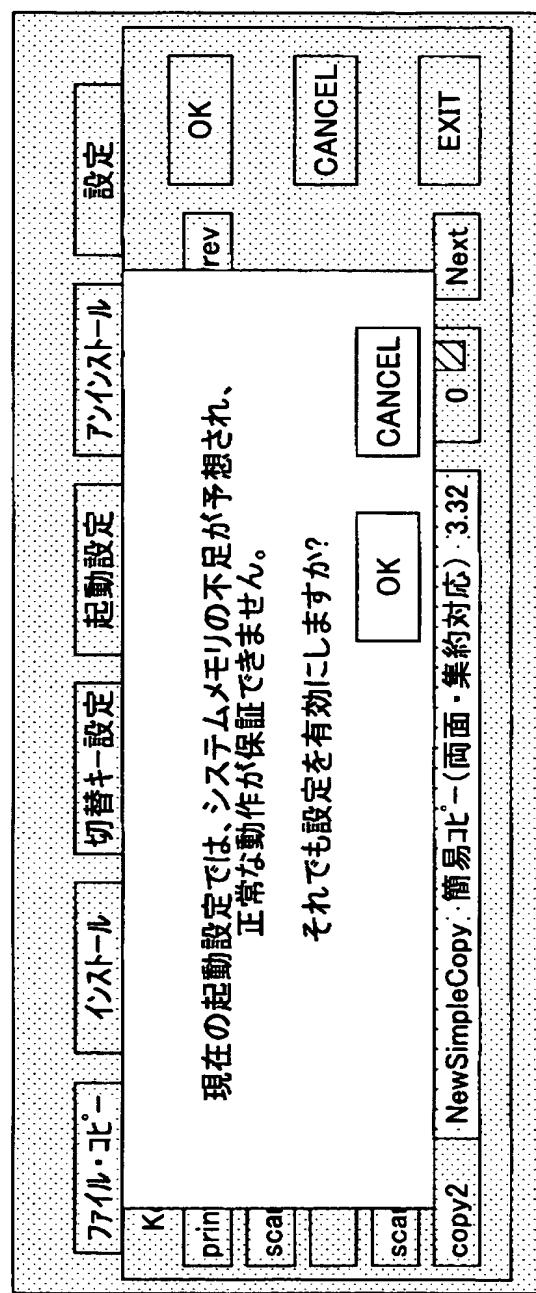
【図 9】

他のアプリケーションの選択を促す画面



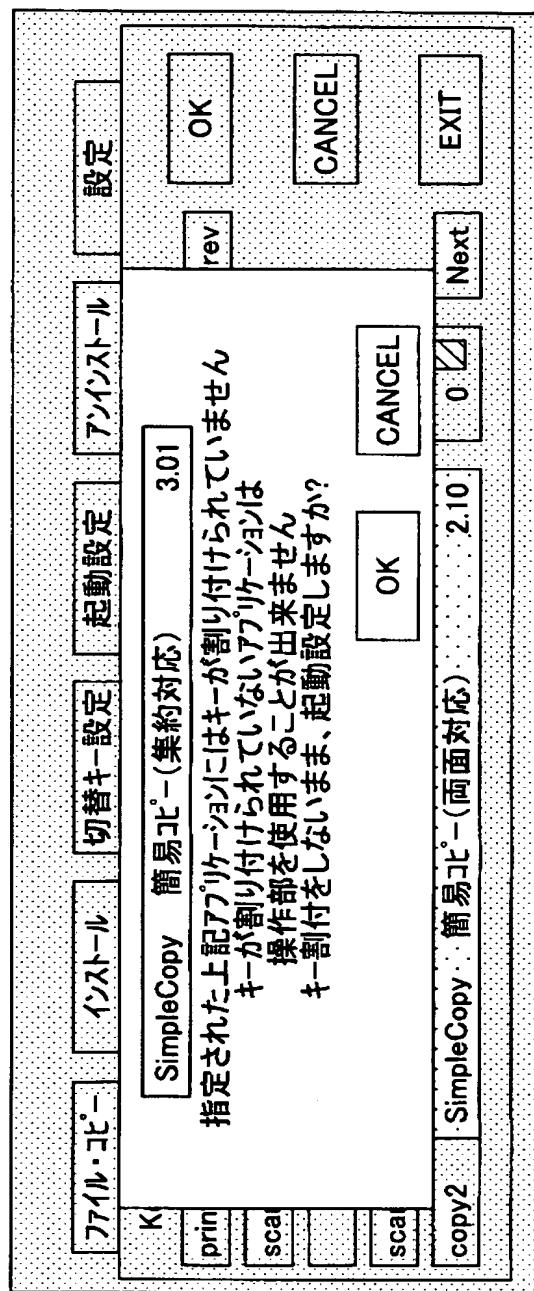
【図10】

メモリ不足でも起動設定を可能とする画面の例



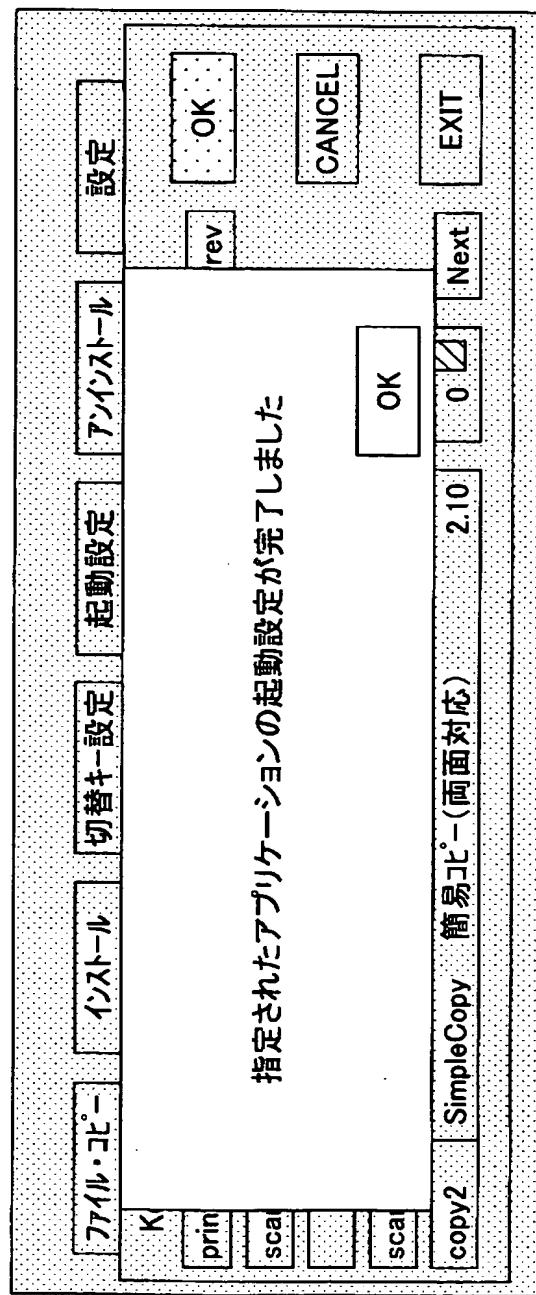
【図11】

キーが割り付けられていないアプリケーションを選択した場合の画面



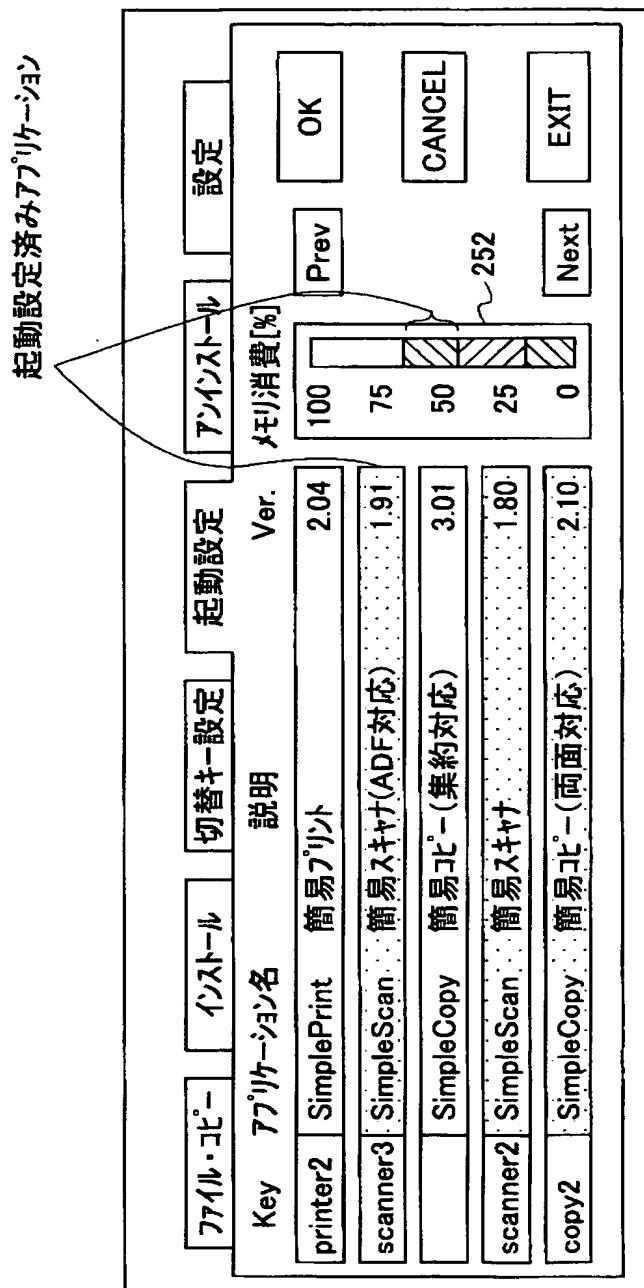
【図12】

起動設定が完了したことを示す画面



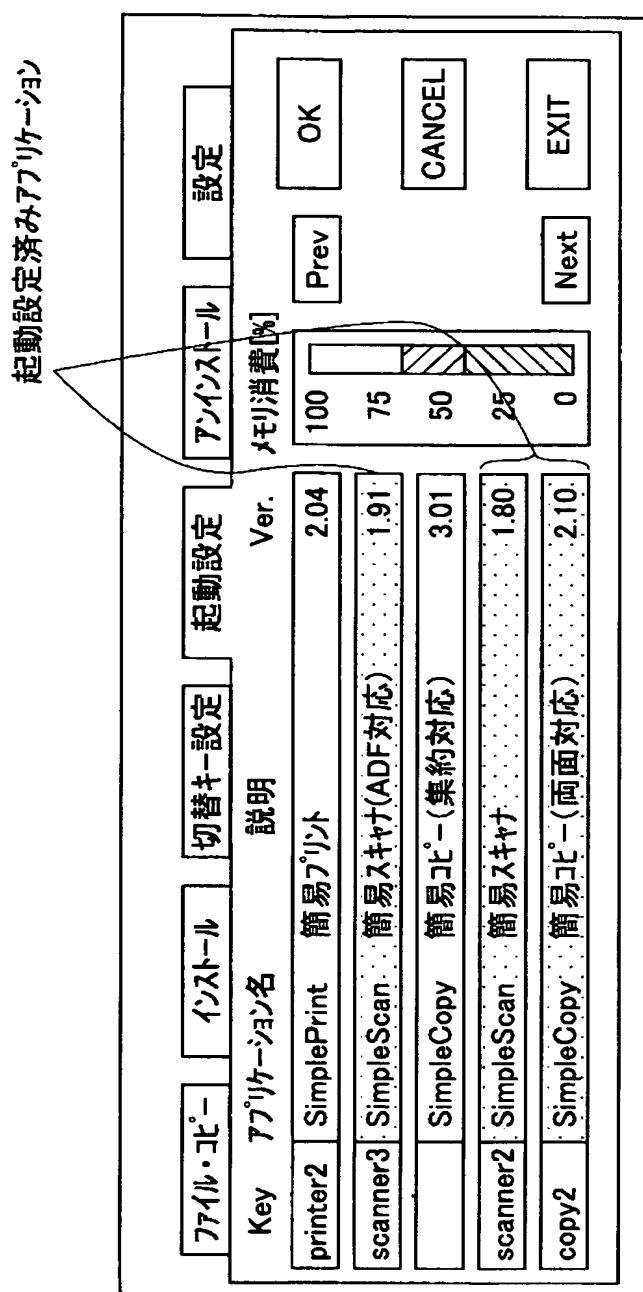
【図13】

「簡易スキャナ」の起動設定が完了した後の画面



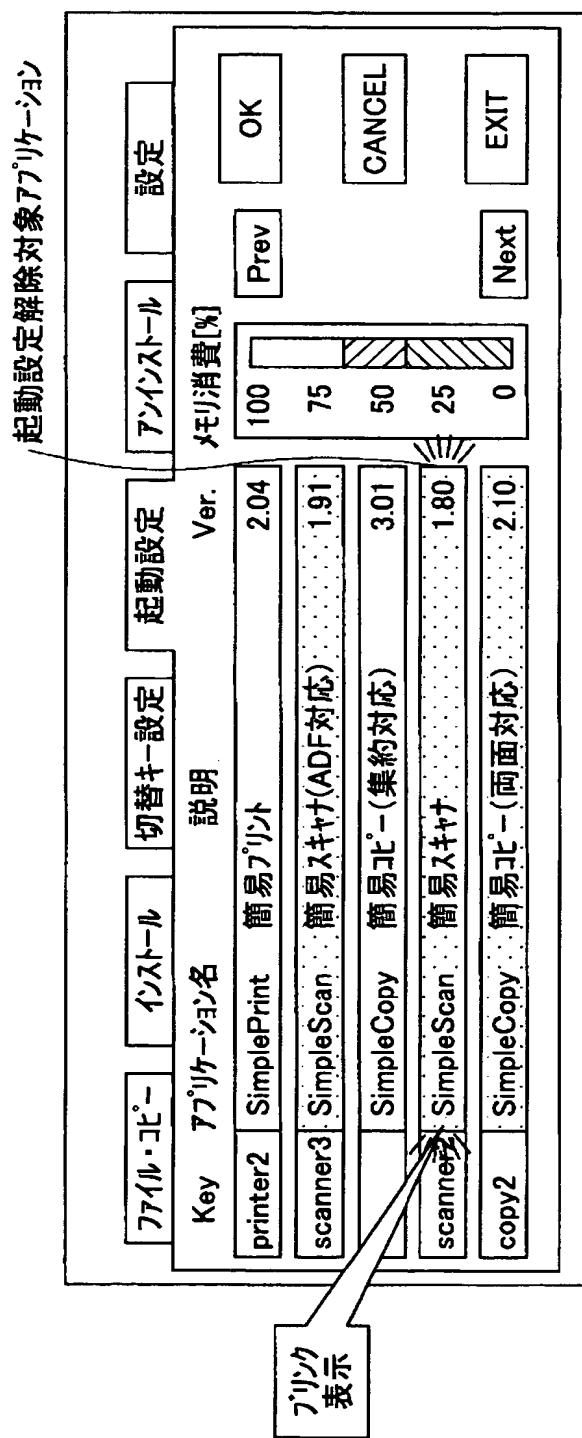
【図 14】

起動設定済みのアプリケーションを示す画面



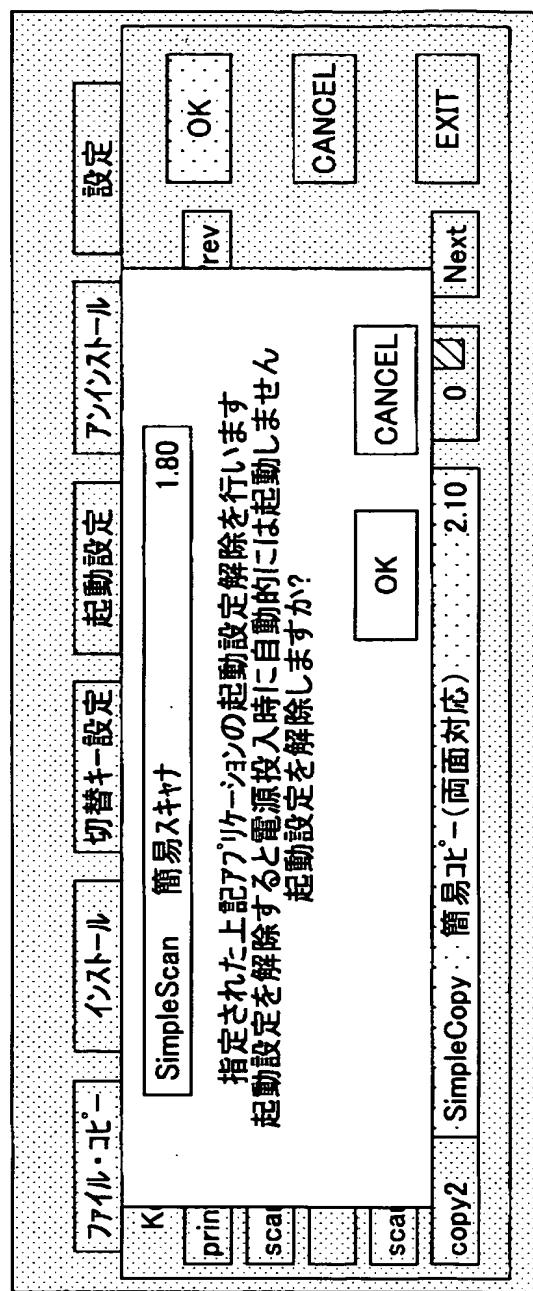
【図15】

「簡易スキャナ」を選択した場合の画面



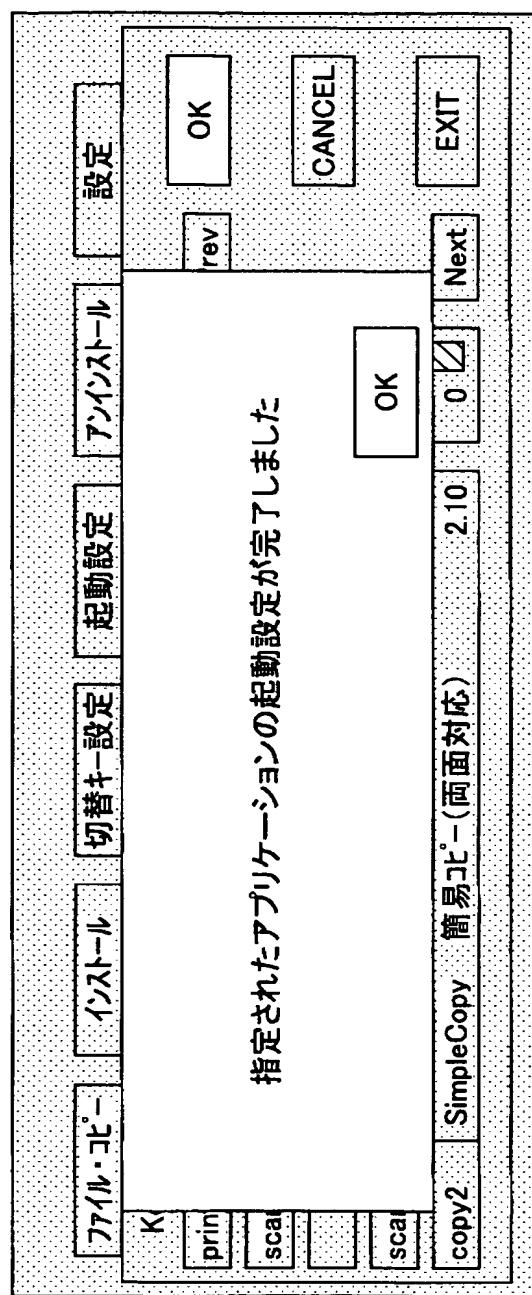
【図16】

起動設定解除を確認する画面



【図17】

解除が完了したことを示す画面



【図18】

解除後の画面

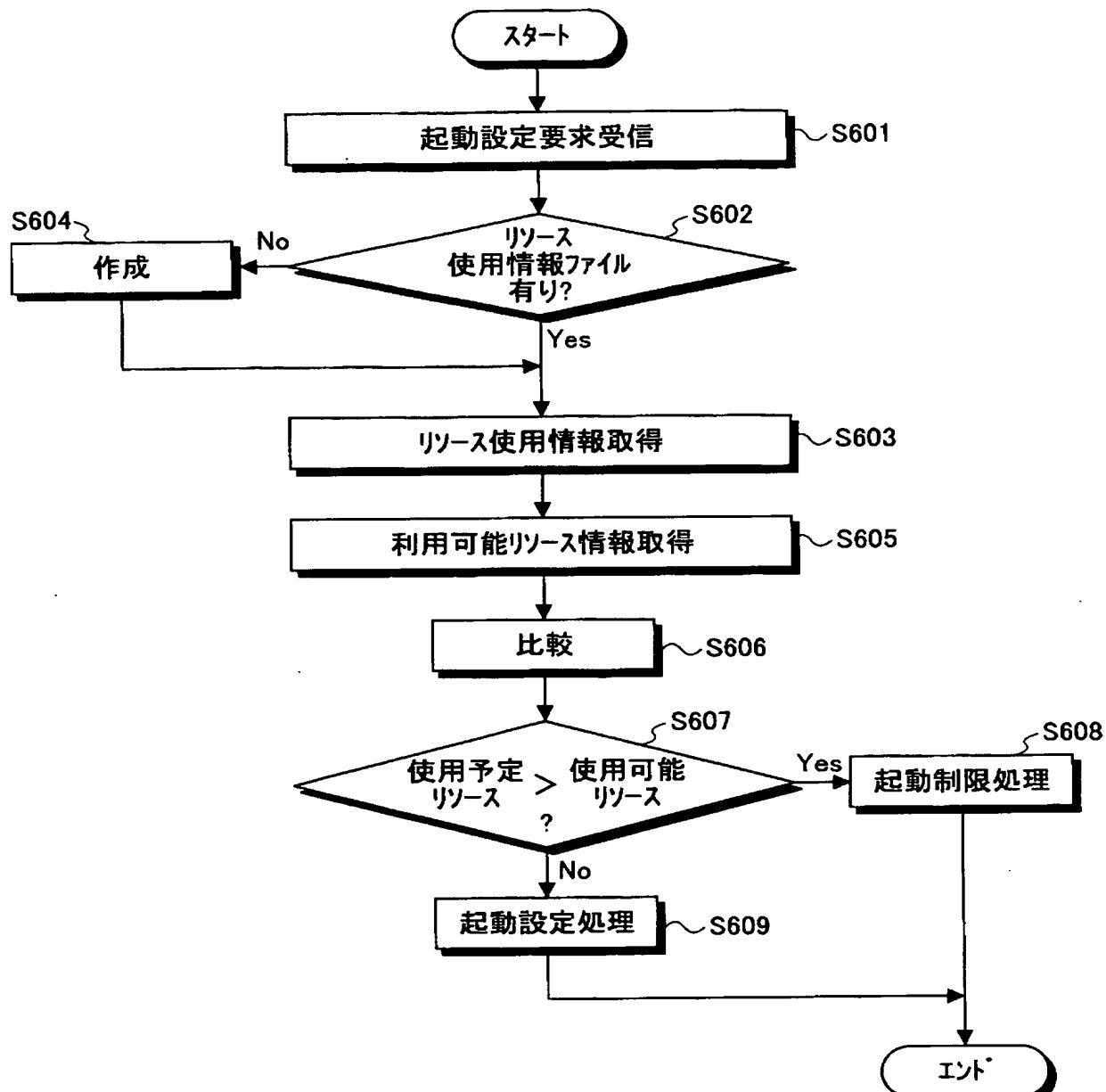
起動設定済みアプリケーション

ファイル・コピー	インストール	切替キー設定	起動設定	アンインストール	設定
Key アプリケーション名	説明			メモリ消費[%]	
printer2 SimplePrint	簡易プリント		Ver. 2.04	100	Prev
scanner3 SimpleScan	簡易スキャナ(ADF対応)			75	OK
				50	CANCEL
				25	
				0	Next
					EXIT

初期設定画面へ

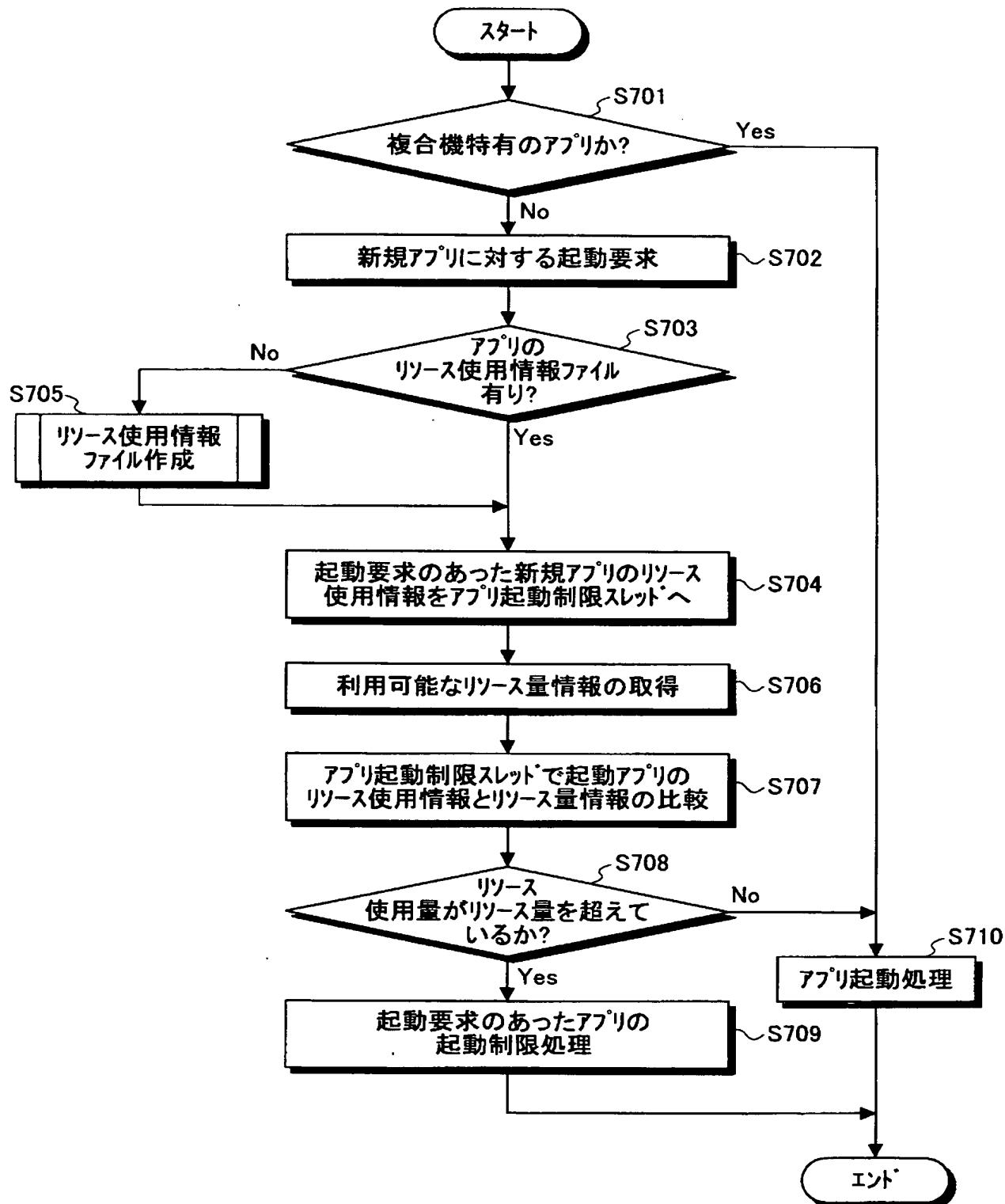
【図19】

起動設定時における起動制限処理手順を示すフローチャート



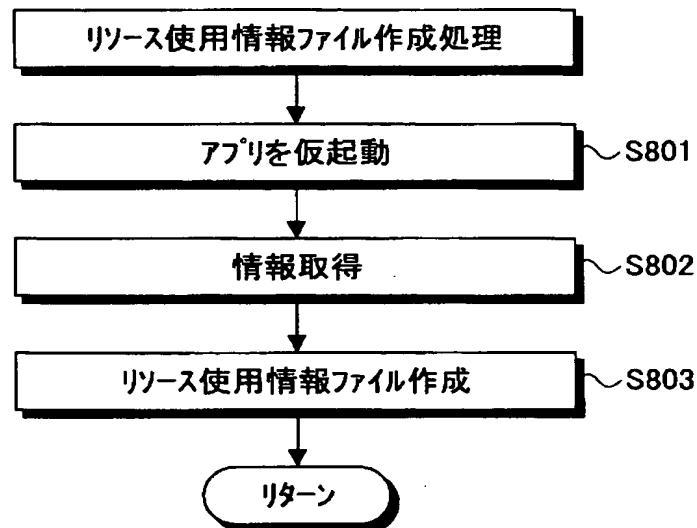
【図20】

起動時における起動制限処理手順を示すフローチャート



【図 2 1】

リソース使用情報ファイルの作成処理のフローチャート



【図 2 2】

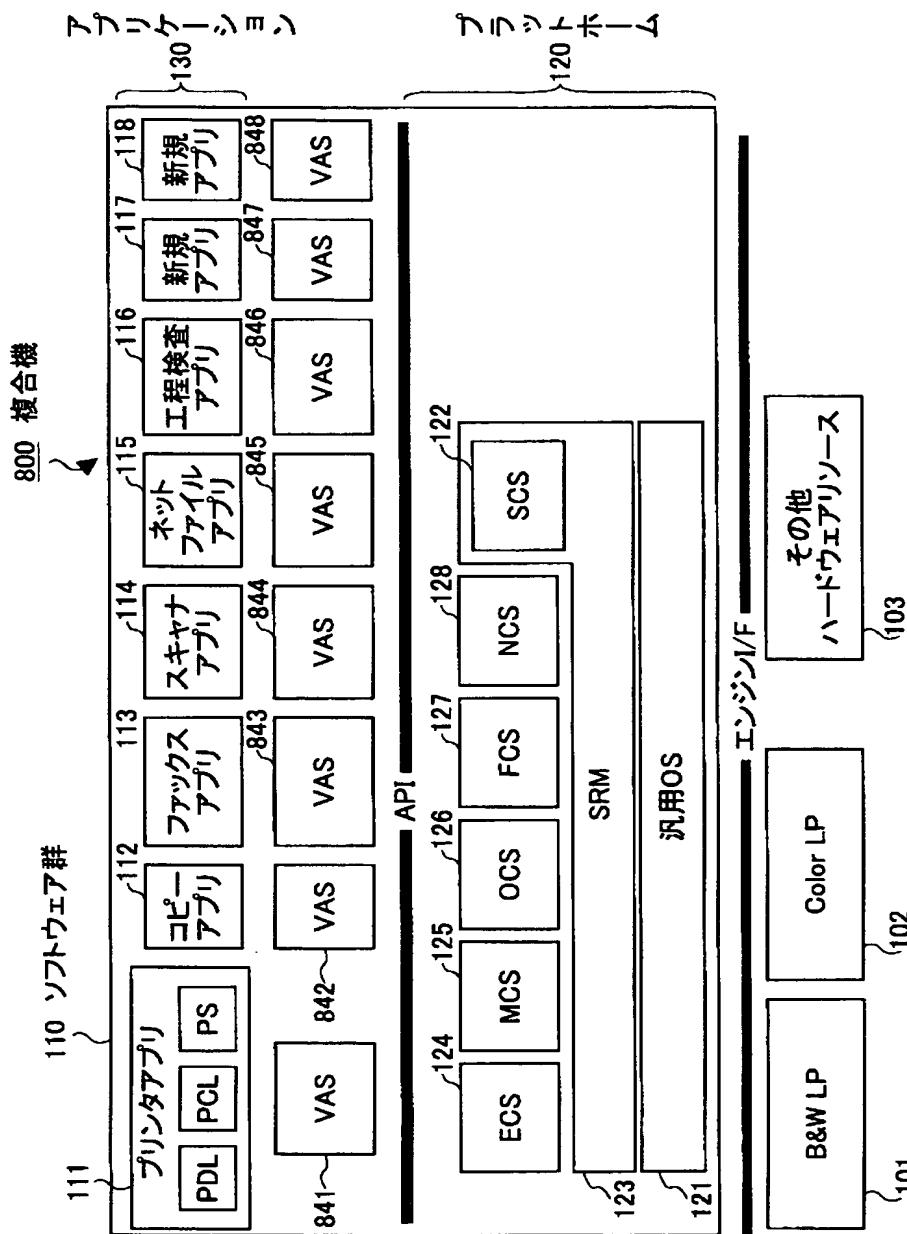
仮起動の機能を含むアプリのプログラム記述
(メイン関数)の概要を示す図

```

int main(int argc, char** argv)
{
    if(argc==2){
        if(strcmp(argv[1], "-v")==0)
            //仮起動実行
            if(ConnectVAS()==OK){ //VASとの通信準備
                SetApplInfo(); //アプリ情報提供処理
                Close //通信終了
            }
            exit(1)
    }
    //通常起動処理
}
  
```

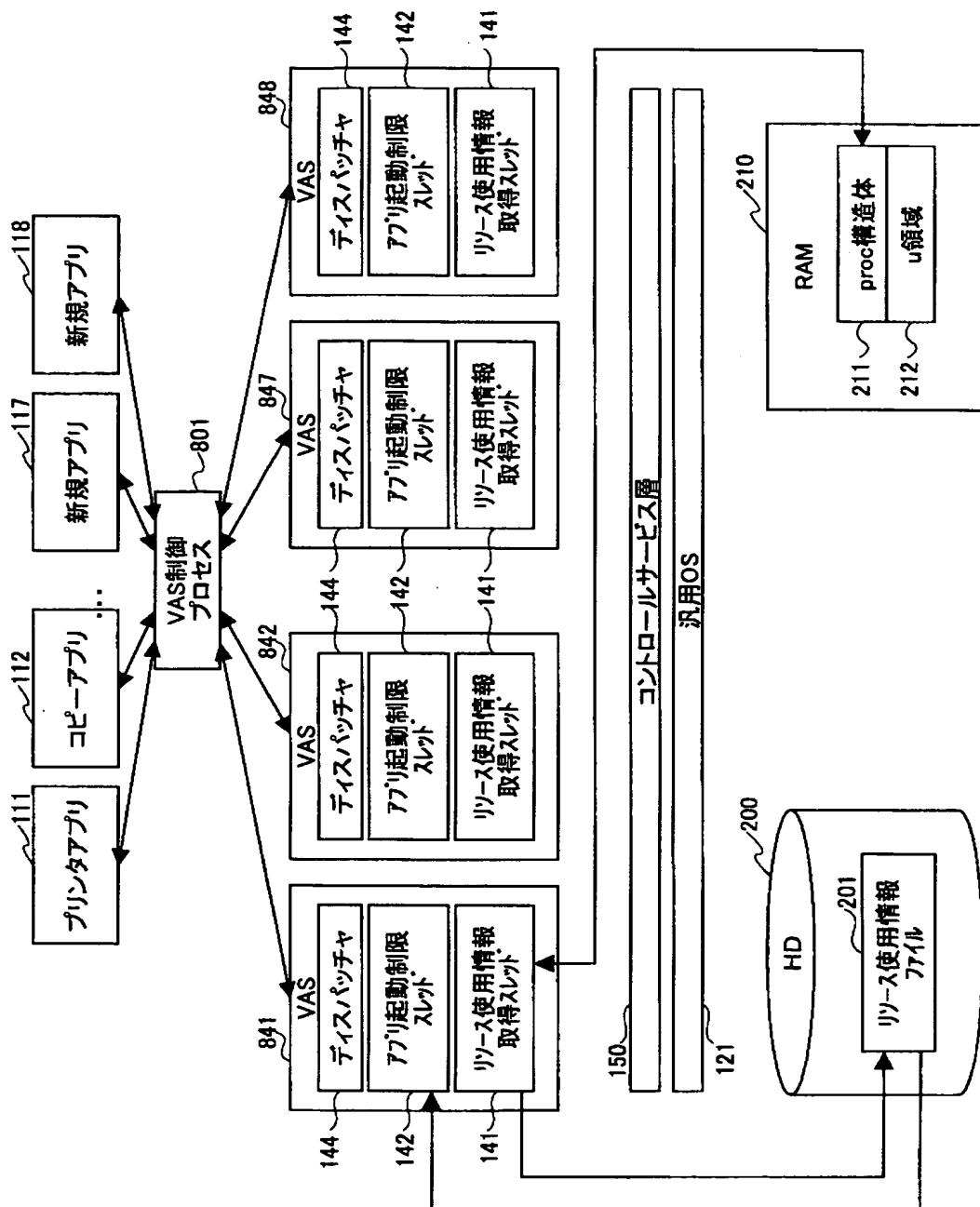
【図23】

実施の形態2にかかる複合機の構成を示すブロック図



【図24】

実施の形態2にかかる複合機のVASの構成と、
VASと各アプリ、コントロールサービス層および
汎用OSとの関係を示すブロック図



【書類名】要約書

【要約】

【課題】 リソースの状況に基づき、アプリケーションを起動制限することを可能とする

【解決手段】 複数のアプリケーションを搭載可能に構成された画像形成装置において、あるアプリケーションを実行するために必要な必要リソースの情報と、前記画像形成装置において使用できる使用可能リソースの情報を取得する情報取得手段と、前記必要リソースの情報と前記使用可能リソースの情報を比較し、比較の結果に基づき、前記アプリケーションに対する起動制限処理を行う起動制限手段とを備える。

【選択図】 図 6

特願 2003-317280

出願人履歴情報

識別番号 [000006747]

1. 変更年月日 2002年 5月17日

[変更理由] 住所変更

住 所 東京都大田区中馬込1丁目3番6号
氏 名 株式会社リコー